MULTI-AGENT COORDINATION: FLUID-INSPIRED AND OPTIMAL CONTROL APPROACHES

A Thesis
Presented to
The Academic Faculty

by

Peter M. Kingston

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2012

MULTI-AGENT COORDINATION: FLUID-INSPIRED AND OPTIMAL CONTROL APPROACHES

Approved by:

Dr. Magnus Egerstedt, Advisor
School of ECE
Georgia Institute of Technology

Dr. Erik Verriest
School of ECE
Georgia Institute of Technology

Dr. Santiago Grijalva
School of ECE
Georgia Institute of Technology

Dr. Anthony Yezzi
School of ECE
Georgia Institute of Technology

Dr. Panagiotis Tsiotras
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: 28 March 2012

ACKNOWLEDGEMENTS

Thanks must go, first and foremost, to my advisor, Dr. Magnus Egerstedt – for his support during my Ph.D.; for creating a uniquely positive academic environment and research group; and for giving me the freedom to do interesting and varied research. At all stages, Dr. Egerstedt has been an unfailing advocate, and for this I cannot thank him enough. I would also like to thank my Ph.D. committee, Dr. Santiago Grijalva, Dr. Panagiotis Tsiotras, Dr. Erik Verriest, and Dr. Anthony Yezzi. I appreciate tremendously the time and effort that they have taken to review this dissertation. Additionally, this work would not have been possible without the generous support of the Office of Naval Research and the National Science Foundation.

I would like to thank Dr. Todd Murphy, who was kind enough to host me at Northwestern University and to introduce me to Geometric Mechanics. Ideas that I first encountered in Chicago have illuminated my research in often-unexpected ways. I would also like to warmly acknowledge Dr. Mike Stilman and Tobias Kunz, with whom I enjoyed an excellent collaboration.

My present and former colleagues at the Georgia Robotics and Intelligent Systems (GRITS) Lab have been an unrivaled source of insight and comradeship. GRITS Lab: For our discussions (technical and otherwise), for your advice, and for your friendship, I count myself lucky. It has been a singular privilege to work with you.

Finally, thank you to my family – for encouraging me to begin this journey, and for supporting me through its completion.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

Multiagent coordination problems arise in a variety of applications, from satellite constellations and formation flight, to air traffic control and unmanned vehicle teams. We investigate the coordination of mobile agents using two kinds of approaches. In the first, which takes its inspiration from fluid dynamics and algebraic topology, control authority is split between mobile agents and a network of static infrastructure nodes - like wireless base stations or air traffic control towers - and controllers are developed that distribute their computation throughout this network. In the second, we look at networks of interconnected mechanical systems, and develop novel optimal control algorithms, which involve the computation of optimal deformations of time- and output- spaces, to achieve approximate formation tracking. Finally, we investigate algorithms that optimize these controllers to meet subjective criteria of humans.

CHAPTER I

INTRODUCTION

Multiagent coordination problems arise in a variety of applications, including satellite constellations, formation flight, air traffic control, and unmanned vehicle teams. Motivations are multifold: to produce systems that are robust to the loss of individual components; to position instruments at scales at which mechanical structures are impractical, e.g., for interferometry, phased-array applications, or multi-view imaging; to enable existing systems like military platforms, by coordinating, to collectively solve larger problems than those for which they were originally designed; and to impose order on systems, like electrical grids and transportation networks, that arise from distributed social structures. Much of the inspiration for multiagent approaches comes from biology, where flocks, schools, and packs of animals coordinate to solve problems – like flying with aerodynamic efficiency, foiling predators, or hunting prey – but physics also has a role to play, where ensembles of particles, like those comprising fluids or electrical conductors, exhibit properties that arise only from their interaction.

Despite the immense growth in interest in deploying multi-agent and swarm robotic solutions, and despite the intuitive biological and physical justifications for doings so, there remains a need for *useful abstractions* for controlling these systems. The goal of this dissertation is to fill this gap, by presenting *novel abstractions* for multiagent systems.

We will focus on two complementary, physically-inspired approaches to multiagent coordination. To describe them, we will borrow the terms *Eulerian* and *Lagrangian* from fluid dynamics. The *Eulerian* approach considers points or control volumes that

are fixed in space, and the dynamics one obtains relate values that are assigned to these points – like density, temperature, and pressure – to flows of mass, energy, and force. This is to be contrasted to the *Lagrangian* approach, in which coordinates are employed that move *with* the fluid – which has the interpretation of tracking an *individual particle* of the fluid. For our purposes, Eulerian control schemes for multiagent systems will be those that work with fixed control volumes containing mobile agents, and Lagrangian control schemes will be those that track individual agents, as particles. The key distinction between the two approaches is that, whereas the Lagrangian approach is concerned with *what state each agent is in*, the Eulerian approach only considers *how many agents are in each state*.

1.1 Eulerian Swarms

For implementation, the Eulerian approach to multiagent control splits control authority between mobile agents and a network of static infrastructure nodes – like wireless base stations or air traffic control towers – that are responsible for different regions of the environment. Algorithms are then developed that distribute their computation throughout the *static* network, to produce local controllers that are broadcast to mobile agents within each region – an approach that we first considered in [55].

An underlying assumption for the Eulerian approach is that mobile agents are anonymous and interchangeable. Then, rather than requiring agents to engage in complicated role-assignment negotiations – some of which result in problems that are NP-hard – we exploit the existence of static infrastructure to discard agent identities a priori by representing the joint state of a swarm as a density function over a state space. Indeed, a desire to simplify multiagent algorithms by discarding agent identities is an important motivation for the approach.

Another motivation is the recognition that real multiagent systems often do use

static communications infrastructure, so in these situations, it is natural to ask whether that infrastructure can be used for control as well. Moreover, in certain application domains, like air traffic control, static infrastructure is not only widespread, but indeed has legal authority. The Eulerian approach is compatible with these existing forms of engineering and social organization.

Finally, the distributed-infrastructure approach harnesses a number of extremely interesting tools from algebraic topology, Hodge Theory, and the theory of discrete harmonic functions – which gives, in addition to the practical reasons just outlined, an aesthetic motivation for considering the Eulerian approach.

1.2 Lagrangian Swarms

In a complementary approach, we consider multiagent system representations that do track individual agents' states. In particular, we look at networks of interconnected mechanical systems, and develop novel optimal, approximate formation tracking algorithms. As in the distributed-infrastructure approach, graph-based models are used to represent robot formations; in this case, graph edges represent constraints that couple agents. The chief source of novelty is the development of a number of new optimal control problems, which involve additional degrees of freedom, and their application to these constraint-coupled swarms. Specifically, we allow for both the time axis and the output space to be deformed elastically, and optimize over the homeomorphisms defining these warpings, to achieve approximate formation tracking. We first addressed the time- and output- warped tracking problems in [59] and [61].

Deformation of the time axis is motivated by the recognition that the controlled system either simply may not be able to move as quickly as a human, or may have natural modes of oscillation (imagine waves propagating through a swarm) that can be exploited by interpreting time liberally. In other words, although one system may not be able to execute a motion at the same speed as another, it may nevertheless be able

to perform an "equivalent" motion by moving at a different speed. This approach was originally motivated by problems in robotic puppetry, where, for instance, a marionette actuated by weak servos whose arm swings with a natural frequency of 1 Hz may be asked to mimic a human waving her hand at 2 Hz. By choosing to interpret the human's wave at a slower speed, the oscillatory dynamics of the marionette can be used to create a more convincing wave. In the same way, we aim to allow controllers to harness, rather than fight, the natural modes of swarms. This is enabled by a flexible interpretation of time, which will arise automatically as the solution to one of a number of optimal control problems.

In addition to deforming the time axis, we apply homeomorphisms to the output space as well. Returning to the example of the marionette asked to mimic a waving human, suppose that the human waves at head level yet the puppet cannot raise its hand above shoulder level. Using any of the usual metrics, the marionette simply cannot track this reference signal. However, if the human's "wave" motion is allowed to be translated down to shoulder level, it can be tracked and the puppet can "mimic" the human by executing a motion which is lower but still recognizably the same. In a similar way, swarm formation specifications may be deformed to allow swarms to subjectively track human commands.

1.3 Subjective Controller Tuning

Finally, we investigate methods for using data of humans' preferences, in order to "learn" controllers, under both frameworks, that optimize humans' subjective experience in controlling swarms. Our goal in this complementary work, which we explored in [58], [60], and [57], is not to investigate human psychology itself, but rather to develop the tools necessary for optimal subjective control.

Entertainment and artistic applications help to motivate this aspect of the problem, where robots are asked to generate motions that, rather than achieving a well-defined physical outcome, instead serve an aesthetic or communicative purpose. In these situations, the effectiveness of control is ultimately the degree to which it aligns with the subjective judgements of human observers.

The role of human preferences in these problems is unavoidable, in that a swarm behavior or motion is aesthetically pleasing only if we think it is pleasing. We will address techniques both for using empirical measurements to learn cost functions that are consistent with humans' aesthetic preferences, and for generalizing from these preference measurements to determine a globally best alternative.

Work in this area has the opportunity to impact areas beyond control and robotics, with applications ranging from recommendation systems as used by shopping (Amazon, eBay), Internet radio (Pandora, Grooveshark), and advertising (Facebook, Google) websites (see e.g. [44], [1], [74]); to more fundamental study of human behavior, e.g. in psychometrics (e.g., [97], [17]) and economics (e.g., [2], [103], [84]). Nevertheless, for the purposes of this dissertation, preference learning plays a secondary, complementary role to the primary contributions of Eulerian and Lagrangian swarm control schemes.

CHAPTER II

PROBLEM BACKGROUND

Graph theory and algebraic topology provide a natural mathematical language for discussing multiagent coordination. At heart, both simply model *how things are connected*, and for us they will serve two related purposes.

The first is to describe **interactions** between agents. Interactions can take various forms: Agents may communicate with one another over traditional communications channels (e.g., radio, optical, or acoustic links); they may react to one another's physical presence using sensors (like sonar, laser, or infrared rangefinders); or they may mechanically alter one another's states (e.g., by pushing or pulling). The simplest interactions are *pairwise*, and are naturally represented as *edges* of a graph that takes the agents as its nodes, but more complicated interactions involving three or more agents can also be modeled using a generalization of graphs, the *abstract simplicial complex*. Graphs and simplicial complexes, which are illustrated in Figure 2, will be introduced formally in the subsequent sections.

The second purpose of graphs and simplicial complexes, for us, is to serve as **spatial abstractions** – both of environments in which agents operate, and of more abstract *output spaces* of systems. In this context, the nodes of graphs, and the

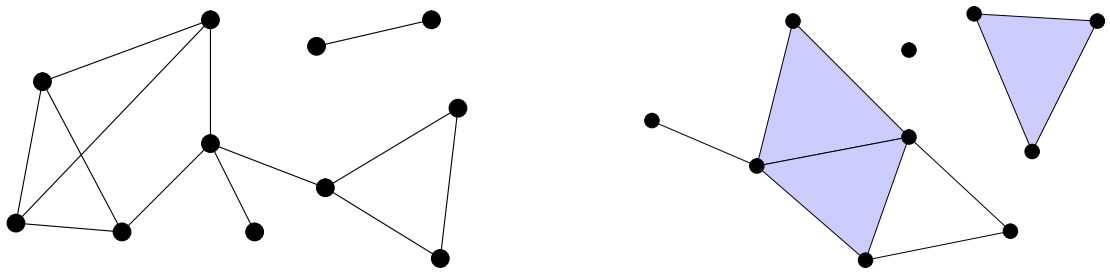

Figure 1: Graph (left), and simplicial 2-complex (right)

triangles, tetrahedra, and higher-order *simplices* of *simplicial complexes*, will represent *regions* of a geometric space. In the *distributed-infrastructure* approach (to be introduced in Section 3.2), abstract simplicial complexes will serve both roles simultaneously – of modeling agent interactions, and of discretizing space – as simplices represent both static agents, like routers, and regions of an environment under their control authority.

2.1 Preliminaries: Algebraic Topology

A number of tools from algebraic topology and homology theory – most notably the Helmholtz-Hodge decomposition, both of a vector field on a smooth manifold (see e.g. [69]), and of a chain on a simplicial complex (as in [40] or [76]) – will play a key role. The analogy between the discrete and continuous in this context is the subject of *discrete exterior calculus* (discussed in [27]), which has found application in a number of areas including computer graphics (e.g. [98]), image processing and clustering (e.g. [37]), computational physics [81], statistical ranking [50], and multiagent control, including [36] where a connection to continuous PDEs is made, [110] which explores a related Laplacian-like operator, [75], which uses higher-order Laplacian dynamics to probe the homology of the complex, and [92], which additionally gives subgradient algorithms to find sparse representatives of the homology groups.

The formalism used in this section closely parallels that of [75] and [92]. Philosophically, however, the goals are very different – in [75] and [92], one seeks to locate holes in a network; here, we will look to understand human preferences and to direct agents throughout an environment. A number of technical differences will also arise from these different goals.

2.1.1 Abstract Simplicial Complexes

We now introduce abstract simplicial complexes more formally. Our exposition will be brief, intended mainly to introduce notation and terminology; for more detail, the

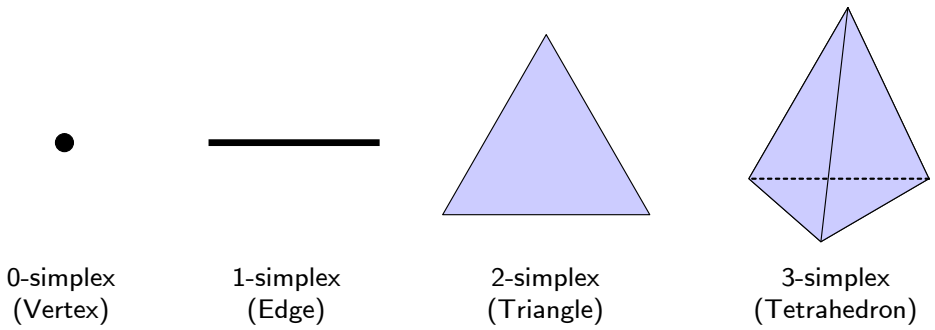

Figure 2: Simplices of various dimension.

interested reader may wish to refer to [40] or [76] (although the formal definitions used in each are slightly different), as well as the introductions to [50] (which uses a dual formulation) and [92]. The definitions that follow in this section are more-or-less standard.

Given a finite set $V(K)$ of *vertices*, a *simplex* $\Delta \subset V(K)$ is a subset of $V(K)$. If the cardinality of that subset is $k+1$, then the *order* of Δ is said to be k, and it is called a k-simplex; simplices of various orders are illustrated in Figure 2.1.1. Any $(k-1)$-simplex $\sigma \subset \Delta$ is a *face* of Δ. A *simplicial complex* K is a finite set of simplices that is closed with respect to taking faces; i.e., if $\Delta \in K$ and σ is a face of Δ, then $\sigma \in K$. A simplicial k-complex K is said to be *pure* if all simplices whose order is less than k are faces of higher-order simplices. We denote the k-simplices of K by $\Sigma_k(K)$. A simplex $\Delta \in K$ is a *coface* of $\sigma \in K$ if σ is a face of Δ. Two simplices σ_1, σ_2 are *lower-adjacent* (denoted $\sigma_1 \smile \sigma_2$) if they share a face, and *upper-adjacent* (denoted $\sigma_1 \frown \sigma_2$) if they share a coface. A simplicial complex is illustrated in Figure 2.

An *orientation* of a simplex is a total order over its vertices, modulo even permutations, with a formal sign,[1] as is illustrated by Figure 2.1.1. If the set $\Delta = \{v_0, \cdots, v_k\}$ is a simplex, we denote an orientation of Δ by an ordered list, e.g. $[v_0, \cdots, v_k]$. Two orientations related by an odd permutation are said to be *opposite*, and this is written

[1]The formal sign is necessary only to allow 0-simplices to have two orientations.

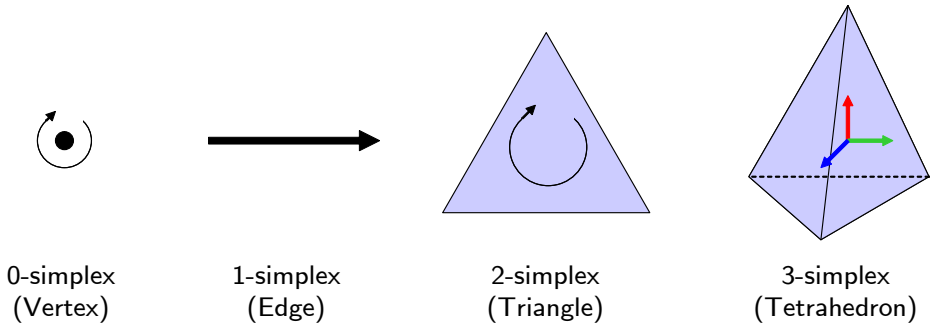

Figure 3: Oriented simplices of various dimension.

with a minus sign; for instance $[v_0, v_1, v_2] = -[v_1, v_0, v_2]$. Finally, an orientation of a simplex induces an orientation on its faces; the i-th oriented face of an oriented simplex $\Delta = [v_0, \cdots, v_k]$ is,

$$\begin{aligned} F_i(\Delta) &= (-1)^i [v_0, \cdots, v_{i-1}, v_{i+1}, \cdots, v_k] \\ &= (-1)^i \Delta / v_i \ . \end{aligned} \tag{1}$$

Likewise, an orientation of a simplicial k-complex is an assignment of an orientation to each of its k-simplices. A simplicial k-complex is *consistently oriented* if, for every pair of lower-adjacent k-simplices Δ_1, Δ_2 sharing a face σ, Δ_1 and Δ_2 induce opposite orientations on σ.

A *k-chain* $c \in C_k(K)$ over an oriented simplicial complex K is a formal sum of elements from $\Sigma_k(K)$ taking coefficients from some commutative ring; we use the real numbers, $\mathbb{R}$. For instance, the formal sum $1.2v_0 + 2.6v_1 - 0.5v_4$ is a 0-chain over an appropriate simplicial complex. Formal sums can be added and multiplied by scalars in the natural way, so $C_k(K)$ forms a finite-dimensional real vector space. Additionally, we equip $C_k(K)$ with an inner product, $\langle \cdot, \cdot \rangle$, defined by

$$\left\langle \sum_{i=0}^{N} a_i \sigma_i, \sum_{i=0}^{N} b_i \sigma_i \right\rangle = \sum_{i=0}^{N} a_i b_i \tag{2}$$

where $\Sigma_k(K) = \{\sigma_0, \cdots, \sigma_N\}$, and $a_i, b_i \in \mathbb{R}\ \forall i$ are the chain coefficients.

Boundary operators will be central to this work. The k-th boundary operator

$\delta_k(K) : C_k(K) \to C_{k-1}(K)$ on the oriented simplicial complex K is defined,

$$\delta_k(K)\left(\sum_{i=0}^{N} a_i\sigma_i\right) = \sum_{i=0}^{N} a_i \sum_{j=0}^{k} F_j(\sigma_i) \; ; \tag{3}$$

by convention, $\delta_0(K) = 0$. The null space of $\delta_k(K)$ is called the *k-cycles* of K and denoted $Z_k(K)$; the image of $\delta_{k+1}(K)$ is called the *k-boundaries* and denoted $B_k(K)$. The k-th homology group is the quotient space $H_k(K) = Z_k(K)/B_k(K)$; its dimension is the *k-th Betti Number* of K.

The *k-th combinatorial Laplacian* is defined, $\mathcal{L}_k(K) = \delta_k^*(K)\delta_k(K)+\delta_{k+1}(K)\delta_{k+1}^*(K)$, where $\delta_k^*(K)$ denotes the adjoint operator to $\delta_k(K)$, called the *k-th coboundary operator*. The matrix representations of $\delta_k(K)$ and $\delta_k^*(K)$ are transposes of one another.

For the special case when K is 1-dimensional and so isomorphic to a graph, we may also use terminology from graph theory [28].[2] There, $C_0(K)$ is called the *vertex space*, $C_1(K)$ is the *edge space*, $\delta_1(K)$ is the *cycle space* (and its dimension is the *cyclomatic number*), and image $\delta_1^*(K)$ is the *cut space*.

A *realization* of a simplicial complex K is an isomorphic complex K' whose vertex set $V(K')$ is a finite subset of $\mathbb{R}^n$ for some $n \in \mathbb{N}$, and its *Rips Shadow* $\mathcal{R}(K') \subset \mathbb{R}^n$ is the union of the convex hulls of its simplices' vertex sets.

2.2 Index-Free Multiagent Systems

The systems that we will be controlling are swarms of interchangeable agents. Consequently, we are interested in problem formulations that do not require agent identities.

In much of the multiagent systems literature, it is assumed that there is an indexed collection of agents with states $x_1, \cdots, x_N \in \mathbb{R}^n$. Then, these individual states are collected in a single vector $x \in \mathbb{R}^{nN}$, and analysis proceeds from this point. Examples include [78], [107], [32], [94], [11], [12], [85], [104], [30], [31], [39], [71], [79], [95], [96], [106], [108], [56]. For systems in which there are many homogeneous agents, however,

[2] In graph theory, it is more common to use the two-element field $F_2 = \{0, 1\}$ (i.e., XOR serves as the addition operation) instead of $\mathbb{R}$.

it may be argued that this is an unnatural approach because it requires that agents be indexed, whereas the fundamental properties of such systems should be independent of agent labeling. One way of dealing with this is to introduce an equivalence relation between states, defined

$$x \sim y \iff \exists P \in \pi(N) \text{ s.t. } x = (P \otimes I)y \tag{4}$$

for any $x, y \in \mathbb{R}^{nN}$, where $\pi(N)$ denotes the group of $N \times N$ permutation matrices and $\otimes$ denotes the Kronecker product. Essentially, one is then looking for properties of the system that are invariant under permutation of the agent indices or for methods by which the agents can agree on a permutation; this is the approach taken in e.g. [49], [100], [109].

We will be interested, instead, in swarm representations that are inherently permutation-invariant; this is addressed later in Section 3.1.

Similarly-motivated previous work includes [63] and [62], in which the individual configurations in $\mathbb{R}^2 \cong \mathbb{C}$ of robots in a formation are represented by the roots of a complex polynomial whose coefficients constitute the permutation-invariant joint configuration. Particularly attractive properties of this representation include that it is finite-dimensional and can be interpolated in a very straightforward way. A limitation which we wish to avoid is that it is only applicable to states/configurations in two dimensions, and since the representation hinges on the Fundamental Theorem of Algebra it is not clear how one would generalize to higher (or lower) dimensions.

The work with the largest technical similarities to ours exists in the mathematical biology community under the heading of nonlocal integro-differential models of swarms; this includes [73], [99], [68], [13], and [19]. The models studied in this area typically also include diffusion and sometimes nonlinear reaction terms, but share the property of determining advection velocities by a convolution integral; hence many of the results given in e.g. [99] can be specialized to the case which we will study. Nevertheless, there are a number of essential novelties to our approach which arise precisely

because of the interplay between the indicator distribution and the indexed representation (the latter of which is not present in a purely continuum model). Among these are a natural permutation-invariant geometric structure which the indexed representation inherits via the *kernel trick*, and an Eulerian control philosophy which has the potential to result in dramatically simple controllers, which are the subject of Section 3.1.

2.3 Time and Output Warped Formation Tracking

In this section, we discuss methods that assert, a priori, that a multiagent formation is similar to a given reference if it can be easily deformed or "warped" to match that reference. Specifically, we will consider problems related to the following physical analogy: Suppose one were to draw, using a pencil, the graphs of two signals (representing formation trajectories) on sheets of rubber, and to then stretch those pieces of rubber until the graphs approximately "lined up." Formally, such a process could be described by the pre- or post- composition of the signals with homeomorphisms – an idea that is investigated in more detail in Section 4.2. In preparation for that formal problem statement, in this section we review literature that approaches the comparison of signals or the control of systems using similar ideas.

2.3.1 Time Warping

For the case when control is not the objective but instead one simply wants to compare signals, the *time warping* problem has been studied extensively in the automatic speech recognition literature. There, variations in speaking rate are accounted for by "normalizing" the time axes of different signals. That problem is usually solved in a discrete-time framework with a dynamic programming algorithm known as Dynamic Time Warping (DTW) (for example, see [86], [87], [20], [105]). The research presented in Section 4.2 is inspired in large part by that work, but differs crucially in that we are now interested in the *control* of systems, and *the control problem and the "warping"*

problems are inextricably coupled. What this means is that one cannot simply perform dynamic time warping on a reference signal and then control the system to track it; rather, the two problems must be solved simultaneously.

A number of different but related problems have also been studied in the controls context, as *time reparametrization* (e.g., [80]) or *path following* (e.g. [41],[88],[5]). In [80], for instance, infeasible reference trajectories for feedback-linearizable systems with bounded control inputs are made feasible by reparametrization. Certain minimum-time optimal control problems are introduced, and necessary conditions for the existence of time warping functions are given for this large class of systems. Exact and asymptotic tracking are considered, but finite-time approximate tracking – which we will consider – is not.

Of the previous work, that most similar in approach to our own appears in [83], in which the authors discuss a number of time warping problems in which dynamical constraints need to be enforced. The primary concern of [83] is the comparison of motions for computer vision purposes rather than the control of systems, and this does however motivate different problem formulations and solutions. Linear time-invariant systems are studied; a combination of dynamic time warping and deconvolution is used to solve one formulation efficiently, and another formulation results in an optimal control problem. Key differences from our work include that in [83] time warping is applied to the input to the systems rather than the output as in our case, and in [83] the control effort required to effect the motion is not penalized in the optimization problem. These differences follow naturally from the different goals.

2.3.2 Output Warping

Conceptually, output warping is inspired in large part by work in "elastic" image deformation, which has applications to image registration (e.g., [54], [67], [70], [82], [101]), video compression (e.g., [93], [45]), and image processing (e.g., [9]). As we will

be "warping" not images but output trajectories, there are however very few technical similarities to our work in this literature (besides the use of simplicial complexes to define warping functions in [9]).

The time- and output- warping problems developed in this dissertation fit naturally into the context of multiagent formations, but can also be applied to fairly general nonlinear systems, including large classes of mechanical systems – e.g., robots like those we have studied in other contexts, like [65]. The warped tracking problems are the subject of Section 4.2.

2.4 Preference Learning

The idea of learning cost or rating functions from expressed preferences has been studied extensively in the machine learning community, where a number of related approaches and problems exist. We sketch a taxonomy of this work here.

In *instance preference learning*, one is given a set of objects called *instances* or *alternatives* (usually points in a real vector space)[3] together with information about humans' preferences among them, and it is sought to learn functions that generalize these expressed preferences to the entire space in some way. When the preference data take the form of values from an ordinal scale – e.g., "Good," "Fair," "Poor" – the problem is known as *ordinal regression* (e.g., [43], [23]). When they take the form of a collection of *pairwise comparisons* (i.e. answers to questions of the form "Which of these two options is better?"), we will refer to the problem as *preference learning* (e.g. [24] [42] [34] [26] [22] [3] [50]). Often, preference learning is done by constructing a real-valued ranking function over the instances (e.g., [34] is representative), but in some cases (particularly when one wishes to allow intransitive preferences), one can seek merely to solve a binary classification problem that determines, for a given pair

[3]To the extent that a distinction is made between "instances" and "alternatives," it is that "instances" are the points that were shown to human judges, whereas "alternatives" may also include other points in the space besides those that were seen.

of instances, which is preferred (as in, e.g., [42]). Applications have included route selection by automotive GPS systems [34], food preference studies [10] [29], and the sorting of search results [51] [26], among many others.

It should be noted that pairwise comparison studies have the advantage over numerical or ordinal-ranking experiments of being less prone to *batch effects*, a psychological phenomenon in which people's rankings are only accurate among objects compared at around the same time [29]. Specific experimental protocols include two-alternative forced choice (2AFC), in which, when comparing two objects A and B, subjects must either respond "A is better than B" or "B is better than A;" three-alternative forced choice (3AFC), in which "the two are the same" is also an acceptable answer; and 4AFC, which also includes "the two are incomparable" as an answer.

The problem of *label preference learning* (e.g. [48] [7]) introduces an extra level of complexity over instance preference learning. Here, two sets of objects are considered, one of *instances* and the other of *labels*; then, one is given, for each instance, a partial order (or *label ranking*) over labels for that instance. In the prototypical example, the instances are documents, the labels are document classifications, and the label ranking for e.g. the book *Mathematical Models in the Social Sciences* [53] would indicate that it is more "Mathematics" than "Social Sciences," and more "Social Sciences" than "Biology." Continuing the example, the goal is to learn from these data a function that, when presented with some unseen book, will return a label ranking for it.

For solving instance- and label- preference problems, large-margin approaches, which ultimately employ Support Vector Machines (SVMs), dominate in the literature; most of these approaches draw heavy inspiration from [43]. However, Bayesian [22] and least-squares [50] approaches also exist.

The least-squares approach of [50] is worth singling out, because a very strong link to combinatorial Hodge theory exists here. Indeed, [50] represents preferences

as 1-chains on a simplicial 2-complex, and considers projections of these preference-representing 1-chains onto conservative and harmonic subspaces.

Our work, to be presented in Chapter 5, falls into the category of large-margin instance preference learning for 2- or 3- AFC experiments, but differs from existing literature in that the goal of computing a globally-optimal alternative motivates different problem formulations and optimization problems, which allow for the efficient determination of that alternative. Ultimately, the role of preference learning in this dissertation is as a complementary technology – as a method to tune the controllers, designed using the Eulerian and Lagrangian control methodologies that are the emphasis of our work, so that they align with humans' subjective preferences.

CHAPTER III

EULERIAN SWARMS

In this chapter, we will be interested in describing multiagent systems in ways that discard agent identities, and, often, that offload control authority from individual, mobile agents, to an instrumented environment in which they operate. We will begin this chapter by discussing the index-free multiagent representation itself (first introduced in Section 2.2), and demonstrating that, within this framework, many of the results typically obtained in a more traditional Lagrangian setting can also be achieved without indexes. With this done, we reach the heart of our work in the next section – a collection of algorithms by which static infrastructure nodes can synthesize local controllers that they pass to mobile agents, and that, despite being arrived at by purely local operations, satisfy desirable global properties.

3.1 Index-Free Multiagent Systems via Mass Density Functions

Our discussion of the index-free multiagent system representation will proceed as follows: After introducing the indicator distribution and its dynamics (Section 3.1.1), we relate this representation to the more standard graph-theoretic one (Section 3.1.2); then we prove conservation properties including stability (Section 3.1.2.1). Following this we construct an inner product space on indicator distributions which enables us to reason geometrically about them (and kernelize the Lagrangian representation) (Section 3.1.3), and finally use a discrete analogue to this representation to give an example both of the dual relationship between the Eulerian and Lagrangian approaches, and of the utility for certain problems of selecting the Eulerian approach (Section 3.1.4). In the interest of clean expressions, we freely drop arguments to

functions throughout this section wherever this should not cause undue confusion.

3.1.1 The Indicator Distribution

Beginning from the classical notion of an indexed set of agents, we will build an indicator distribution, or permutation-invariant state, that retains all necessary information while stripping out agent identities. The essential idea will be to construct an object that tells us not what state each agent is in, but rather how many agents are in each state. The construction is as follows: Starting from an indexed collection of agents with states $x_1, \cdots, x_N \in \mathbb{R}^n$, we build an indicator distribution m over $\mathbb{R}^n$ defined,

$$m(x) = \Phi(x_1, \cdots, x_N)(x) = \sum_{i=1}^{N} \delta(x - x_i) \tag{5}$$

where δ is the Dirac delta distribution on $\mathbb{R}^n$, $\mathcal{T}(\mathbb{R}^n)$ denotes the space of tempered distributions on $\mathbb{R}^n$, and the map $\Phi : \mathbb{R}^{nN} \to \mathcal{T}(\mathbb{R}^n)$ creates m from $x_1, \cdots, x_N$. We use m to denote this distribution because we would like to think of it as the agent "mass distribution." Importantly, notice that although indices were used in the construction of m, it is fundamentally an object that is concerned only with the number of agents in any given state.

One may also think of m as the probability distribution (after normalizing by $\frac{1}{N}$) that answers the question, "If an agent is chosen uniformly at random, what is the probability that that agent is at the state x?"

More generally, we would like to be able to add and subtract distributions so that we have a full vector space structure. Hence we will also consider linear combinations of distributions of the form (5); these take the form

$$x \mapsto \sum_{i=1}^{K} c_i \delta(x - \xi_i) \tag{6}$$

for some $c_1, \cdots, c_K \in \mathbb{R}$, $K \in \mathbb{N}$, and $\xi_1, \cdots, \xi_K \in \mathbb{R}^n$.

Now suppose that, in the classical setting, each agent $i \in \{1, \cdots, N\}$ has state

$x_i \in \mathbb{R}^n$ and dynamics

$$\dot{x}_i = v_i \tag{7}$$

where v_i is our control input, and that moreover v_i is generated by some controller, identical for all agents, that depends only on $m(\cdot, t)$ and not the indexed set of states. In other words (and making time dependence explicit), $v_i(t) = v(x_i(t), m(\cdot, t))\ \forall i \in \{1, \cdots N\}$. Then, the equivalent dynamics for our indicator distribution are given by the *advection equation*

$$\dot{m} = -\operatorname{div}(mv) \tag{8}$$

which holds in a weak sense to be described in the next paragraph. Informally, the meaning of (8) is that the rate at which m decreases at a given point is precisely the divergence of the mass flux vector field mv from around that point.

Since m is defined as a sum of Dirac delta distributions, we must be careful to state exactly what is meant by (8). Formally, the distribution m is defined as a linear functional on smooth, rapidly-decreasing test functions called *Schwartz functions*, and its evaluation at a function $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is denoted $\langle m, f \rangle$. In particular, the Dirac delta distribution is identified with the evaluation functional. Then, by analogy with integration by parts, one defines $\left\langle \frac{\partial m}{\partial x_i}, f \right\rangle \triangleq -\left\langle m, \frac{\partial f}{\partial x_i} \right\rangle$; and multiplication by a scalar function $s : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is defined as $\langle sm, f \rangle \triangleq \langle m, sf \rangle$. With these definitions, what (8) really means is that, for any Schwartz function $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$,

$$-\left\langle m, \dot{f} \right\rangle = \langle m, v \cdot \operatorname{grad} f \rangle - \langle m, f \operatorname{div} v \rangle \ . \tag{9}$$

One sees from (9) that the only classical derivatives actually used in (8) are those of the vector field v and of the test function f. For our purposes, it is sufficient merely to note that, with these definitions, integration by parts and the product rule (for the product of a differentiable function with a tempered distribution) both hold. (For a more detailed discussion of these topics, please see [47].)

3.1.2 Weighted Linear Consensus

In the indexed setting, a particular problem that has received a great deal of attention is that of *distributed averaging*. This is described by the *consensus equation*

$$\dot{x}(t) = -L_w(\mathcal{G}(t))x(t) \tag{10}$$

where $L_w(\mathcal{G}(t))$ is the (possibly weighted) *graph Laplacian* for some undirected interaction graph $\mathcal{G}(t)$ on N vertices, and can be written as the product

$$L_w(\mathcal{G}(t)) = D(t)W(t)D^T(t) \tag{11}$$

where $D(t)$ is the *incidence matrix* for any orientation of $\mathcal{G}(t)$ and $W(t)$ is a diagonal matrix of positive edge weights. Specifically, if $G(t)$ is an orientation of an undirected graph $\mathcal{G}(t)$, then $D(t)$ is the matrix representation of the boundary operator $\delta_1(G(t))$ defined in (3); i.e., if $G(t)$ has n vertices, m edges, and edge set $\{(e_1^1, e_1^2)(t), \cdots, (e_m^1, e_m^2)(t)\}$, then $D(t) \in \mathbb{R}^{n\times m}$ is defined by,

$$D_{ij}(t) = \begin{cases} 1 & \text{if } e_j^1(t) = i \\ -1 & \text{if } e_j^2(t) = i \\ 0 & \text{otherwise} \end{cases} . \tag{12}$$

For a more comprehensive overview, please see [72].

We say that the protocol (10) is *permutation invariant* if both the presence of an edge between two agents, and the weight assigned to an edge, are functions only of the permutation-invariant state (or indicator distribution) $m = \Phi(x)$. In other words, the interaction graph is allowed to depend on the states of the many agents, but not on their identities.

This occurs in a great many cases of interest, including disk graphs, Gabriel and Delaunay graphs, nearest-neighbor graphs, and even situations in which edges are functions of many agents' states (e.g., if a line-of-sight communication link between two agents can be severed by a third agent who gets in the way). Consensus on static

(i.e., constant) interaction graphs, however, is generally not permutation-invariant, since in this case edges are determined not by agents' states but by their identities (i.e., if agent i is to communicate with agent j and not with agent k regardless of the many agents' states, then it requires some way to differentiate between agents j and k). In graph-theoretic terms, what is required for a static interaction graph is that every permutation of the vertex labels be a graph automorphism – a property possessed only by the empty graph and the complete graph. Hence, permutation-invariance will usually involve dynamic, state-dependent graphs.

So long as (10) is permutation-invariant, it is possible to express equivalent dynamics in our index-free framework. In short, the state-dependent vector field v in (8) then takes the form,

$$v = \int_{\zeta \in \mathbb{R}^n} m(\zeta) w(\zeta, x, m)(\zeta - x) d\zeta \tag{13}$$

where $w : \mathbb{R}^n \times \mathbb{R}^n \times \mathcal{T}(\mathbb{R}^n) \to [0, \infty)$ is a positive state-dependent weighting function satisfying the symmetry property $w(\zeta, x, m) = w(x, \zeta, m)$ for all $x, \zeta \in \mathbb{R}^n$ and $m \in \mathcal{T}(\mathbb{R}^n)$. In most cases that arise, w depends only on its first two arguments, since the intensity with which a pair of agents interacts is usually a function only of their two states and not on other agents'. For instance, the unit-disk topology[1] can be encoded by the weighting function

$$w(\zeta, x) = \mathbf{1}_{B_1(\zeta)}(x) \tag{14}$$

where, for a given set S, $\mathbf{1}_S$ denotes the zero-one indicator function for S, and $B_r(\zeta)$ denotes the open ball, centered at ζ, of radius r.

A yet-more-specific form for w which will be of particular interest is

$$w(\zeta, x) = f'\left(\frac{1}{2}||\zeta - x||^2\right) \tag{15}$$

where f' is the derivative of some nondecreasing scalar function $f : \mathbb{R} \to \mathbb{R}$.

[1] I.e., two agents can interact if and only if they are within one unit distance of one another.

Regardless of the particular form of w, the controller (13) then induces the closed-loop dynamics on the indicator distribution m,

$$\dot{m}(x) = -\operatorname{div}\left(m(x)\int_{\zeta\in\mathbb{R}^n} m(\zeta)w(\zeta,x,m)(\zeta-x)d\zeta\right) \ . \tag{16}$$

3.1.2.1 Properties of Index-Free Linear Consensus

In this section we prove center-of-mass conservation and stability properties of the closed-loop system (16). Note that the following theorems hold not only when m is a sum of Dirac deltas as in (5), but also for any positive m which either has compact support or which more generally vanishes at infinity; naturally, this includes smooth density functions.

We note here that similar results to Theorems 1 and 2 are also proven for a closely-related PDE featuring diffusive terms in [99]. We nevertheless include Theorems 1 and 2 both in the interest of completeness, and for their value as instructive specializations to our case.

Theorem 1 (Center of Mass Conservation)**.** *Under the assumption that m vanishes at infinity, the center of mass $\bar{x}(t) \in \mathbb{R}^n$ defined by*

$$\bar{x}^i \triangleq \left\langle m, x^i \right\rangle \tag{17}$$

(where x^i is the ith canonical coordinate function) is constant in time.[2]

Proof : Differentiating the i-th component of $\bar{x}$,

$$\begin{aligned}\frac{\partial}{\partial t}\left\langle m, x^i\right\rangle &= \left\langle \dot{m}, x^i\right\rangle \\ &= \left\langle -\operatorname{div}(mv), x^i\right\rangle \\ &= -\sum_{j=1}^{n}\left\langle \frac{\partial}{\partial x^j}(mv)^j, x^i\right\rangle \ .\end{aligned}$$

[2] *Here, $\langle m, \cdot\rangle$ denotes evaluation of m as a linear functional.*

Since m vanishes at infinity, integration by parts gives that this equals

$$\begin{aligned}&\left\langle (mv)^1, \frac{\partial}{\partial x^1} x^i \right\rangle + \cdots + \left\langle (mv)^n, \frac{\partial}{\partial x^n} x^i \right\rangle \\ &= \left\langle (mv)^1, \delta^{1i} \right\rangle + \cdots + \left\langle (mv)^n, \delta^{ni} \right\rangle\end{aligned}$$

where δ^{ij} denotes the Kronecker delta. As a result,

$$\frac{\partial}{\partial t} \left\langle m, x^i \right\rangle = \left\langle mv^i, \mathbf{1} \right\rangle$$

where $\mathbf{1}$ denotes the constant function that returns $1 \in \mathbb{R}$; this bracket is interpreted as the total mass flux. Expanding this expression, we have

$$\begin{aligned}\left\langle mv^i, \mathbf{1} \right\rangle &= \int_{x \in \mathbb{R}^n} m(x) \int_{\zeta \in \mathbb{R}^n} m(\zeta) w(\zeta, x, m) \\ &\qquad \cdot (\zeta - x) d\zeta dx \\ &= \int_{(x,\zeta) \in \mathbb{R}^{2n}} m(x) m(\zeta) w(\zeta, x, m) \\ &\qquad \cdot (\zeta - x) d(x, \zeta) \ .\end{aligned}$$

We note that the term being integrated is antisymmetric in x and ζ, and hence that the integral is zero. ∎

.

Theorem 2 (Stability)**.** *Let $w : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ take the form (15). Then the closed-loop dynamics (16)*

- *are* stable *if $f'(x) \geq 0 \ \forall x \in [0, \infty)$, and*
- *are* globally asymptotically stable *with equilibrium point $\bar{x}$ defined by (17), if $f'(x) > 0 \ \forall x \in [0, \infty)$*

so long as m vanishes at infinity. Moreover, in the first case, if $f'(x) > 0 \ \forall x \in [0, R)$, then for all $i, j \in \{1, \cdots, N\}$, at equilibrium either $||x_i - x_j|| = 0$ or $||x_i - x_j|| > R$.

Proof : Consider the Lyapunov functional,

$$V(m) = \int_{y\in\mathbb{R}^n}\int_{x\in\mathbb{R}^n} m(x)m(y)f\left(\frac{1}{2}||x-y||^2\right)dxdy\ . \tag{18}$$

Differentiating,

$$\dot{V} = \int_{y\in\mathbb{R}^n}\int_{x\in\mathbb{R}^n} [\dot{m}(x)m(y) + m(x)\dot{m}(y)] \cdot f\left(\frac{1}{2}||x-y||^2\right)dxdy$$

which by symmetry

$$= 2\int_{y\in\mathbb{R}^n}\int_{x\in\mathbb{R}^n} \dot{m}(x)m(y)f\left(\frac{1}{2}||x-y||^2\right)dxdy$$

and substituting in (8)

$$\begin{aligned} &= -2\int_{y\in\mathbb{R}^n}\int_{x\in\mathbb{R}^n} \operatorname*{div}_x(m(x)v(x))m(y) \cdot f\left(\frac{1}{2}||x-y||^2\right)dxdy \\ &= -2\sum_{i=1}^{n}\int_{y\in\mathbb{R}^n}\int_{x\in\mathbb{R}^n} \frac{\partial}{\partial x^i}(m(x)v^i(x))m(y) \cdot f\left(\frac{1}{2}||x-y||^2\right)dxdy\ . \end{aligned}$$

Integrating by parts (and since m vanishes at infinity), this is,

$$-2\sum_{i=1}^{n}\int_{y\in\mathbb{R}^n}\int_{x\in\mathbb{R}^n} (m(x)v^i(x)) \cdot \frac{\partial}{\partial x^i}\left(m(y)f\left(\frac{1}{2}||x-y||^2\right)\right)dxdy$$

which, since $f'(\frac{1}{2}||x-y||^2) = w(x,y)$, simplifies to

$$2\sum_{i=1}^{n}\int_{y\in\mathbb{R}^n}\int_{x\in\mathbb{R}^n} m(x)v^i(x)m(y)w(x,y)(x^i-y^i)dxdy\ .$$

Substituting in (13) we arrive at the separable integral

$$\begin{aligned} 2\sum_{i=1}^{n}\int_{x\in\mathbb{R}^n} m(x)\left(\int_{y\in\mathbb{R}^n} m(y)w(x,y)(x^i-y^i)dy\right) \\ \cdot\left(\int_{\zeta\in\mathbb{R}^n} m(\zeta)w(x,\zeta)(\zeta^i-x^i)d\zeta\right)dx \\ = -2\sum_{i=1}^{n}\int_{x\in\mathbb{R}^n} m(x)||v(x)||^2dx \end{aligned}$$

which is never positive so long as $f' \geq 0$ (thus proving the first statement of the theorem), and always strictly negative provided $f' > 0$ (which, together with the previous theorem, proves the second statement). The third statement likewise follows immediately from LaSalle's invariance principle. ∎

3.1.3 An Inner Product Space via Smoothing

In this section we imbue the state space of indicator distributions with an inner product structure; this will enable us to reason geometrically about multiagent control laws in an index-free way. Essentially, we will smooth indicator distributions by convolving them with an appropriate function (in particular, a Gaussian) to arrive at smooth functions for which the standard L_2 inner product is defined; this is illustrated by figure 4. The reason for choosing Gaussians in particular is that the corresponding convolution operator is invertible. In geometric language, the inner product is constructed as the *pullback* of the standard L_2 inner product under a linear isomorphism.

The smoothing operator is introduced in section 3.1.3.1, which is used to define the inner product in section 3.1.3.2. Finally, in section 3.1.3.3, we use this inner product together with the embedding Φ introduced in (5) to compute these inner products directly from the indexed representation.

3.1.3.1 Smoothed Indicator Distributions

Let $w : \mathbb{R}^n \to \mathbb{R}$ be a Gaussian of the form,

$$w(x) = \exp(-x^T Q x) \tag{19}$$

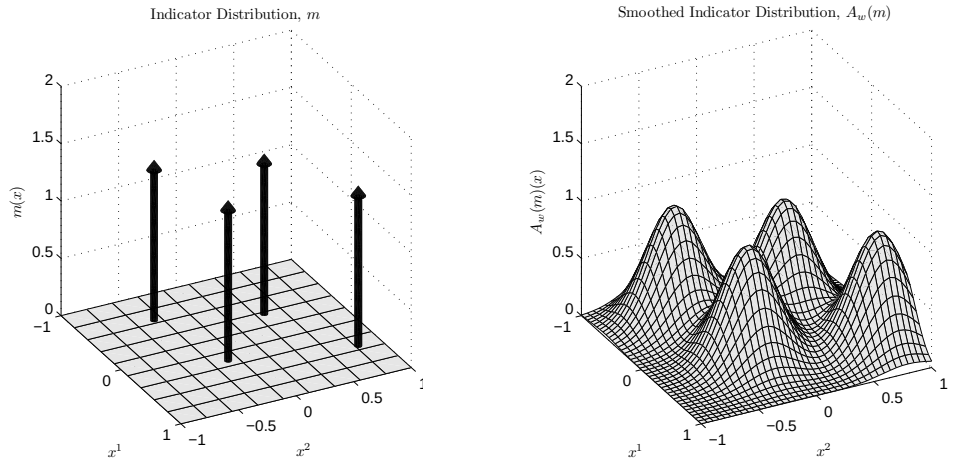

Figure 4: The indicator distribution (left) is smoothed by convolution with a Gaussian to arrive at a function in L_2 (right).

for some $Q = Q^T \succ 0 \in \mathbb{R}^{n\times n}$. Then we define the *smoothing operator* $A_w : \mathcal{T}(\mathbb{R}n) \to L_2(\mathbb{R}^n, \mathbb{R})$ by,

$$A_w(m) = w * m \tag{20}$$

where $*$ is the standard convolution operator.

Lemma 1. *A_w is a linear isomorphism.*

Proof : A_w is clearly linear. To show that it is also invertible, we note that $\tilde{m} = A_w(m)$ can be computed as the solution to Laplace's equation in the following way: The PDE

$$\dot{\phi}(x,t) = \Delta\phi(x,t) \tag{21}$$

with the initial conditions $\phi(z,0) = m(Q^{-1/2}z)$ (where $(Q^{1/2})^T Q^{1/2} = Q$; $Q^{1/2}$ exists since $Q \succ 0$, and can be computed by e.g. the Cholesky decomposition) has the solution

$$\phi(z,t) = \sum_{i=1}^{N} \frac{1}{(4\pi t)^{n/2}} e^{-\frac{(z-Q^{1/2}x_i)^T(z-Q^{1/2}x_i)}{4t}} \tag{22}$$

for $t > 0$, and hence $\tilde{m}(x) = \pi^{n/2}\phi(Q^{1/2}x, \frac{1}{4})$.[3] In the same way, starting with $\tilde{m}$

[3]The function (22) is known as the *heat kernel*.

and imposing the condition $\phi(z, \frac{1}{4}) = \frac{1}{\pi^{n/2}}\tilde{m}(Q^{-1/2}z)$, m can be reconstructed as $m(x) = \phi(Q^{1/2}x, 0)$. ∎

Note that although the backwards heat equation used in lemma 1 is extremely ill-conditioned, in principle the solution exists; see e.g. [33].

Lemma 2. *For any indicator distribution m, $A(m)$ is square integrable.*

Proof : $A(m)$ can be written as a sum of translated copies of w; since w is square-integrable, this sum is square-integrable. ∎

3.1.3.2 An Inner Product

Given two indicator distributions m_1, m_2, we define their inner product

$$\langle m_1, m_2 \rangle_A = \langle A(m_1), A(m_2) \rangle_{L_2(\mathbb{R}^n, \mathbb{R})} \ .$$

Lemma 3. $\langle \cdot, \cdot \rangle_A$ *is an inner product.*

Proof : Since $A(m_1)$ and $A(m_2)$ are square integrable by Lemma 2, their L_2 inner product exists. Since $\langle \cdot, \cdot \rangle_{L_2(\mathbb{R}^n, \mathbb{R})}$ is symmetric, so is $\langle \cdot, \cdot \rangle_A$; since A is linear and $\langle \cdot, \cdot \rangle_{L_2(\mathbb{R}^n, \mathbb{R})}$ is bilinear, $\langle \cdot, \cdot \rangle_A$ is bilinear; and since $\langle \cdot, \cdot \rangle_{L_2(\mathbb{R}^n, \mathbb{R})}$ is positive definite and A is an isomorphism, $\langle \cdot, \cdot \rangle_A$ is positive definite. Hence $\langle \cdot, \cdot \rangle_A$ is an inner product. ∎

3.1.3.3 Kernelizing the Inner Product

How does the inner product of the previous section relate to the classical indexed representation of a multiagent system?

The *kernel* κ_Φ attached to the embedding Φ is the map

$$\kappa((x_1^1, \cdots, x_N^1), (x_1^2, \cdots, x_N^2)) \triangleq$$
$$\left\langle \Phi(x_1^1, \cdots, x_N^1), \Phi(x_1^2, \cdots, x_N^2) \right\rangle \ . \tag{23}$$

In other words, it is the map that computes inner products between joint states in the higher-dimensional space of indicator distributions, without (necessarily) needing

to explicitly construct the indicator distribution representation ([8] has more on the subject of kernel functions).

For indicator distributions of the form (5) and the inner product $\langle \cdot, \cdot \rangle_A$ defined in the previous section, the corresponding kernel is,

$$\begin{aligned}
&\kappa((x_1^1, \cdots, x_N^1), (x_1^2, \cdots, x_N^2)) \\
&= \int_{x \in \mathbb{R}^n} \prod_{i=1}^{2} \sum_{j=1}^{N} A_w(\delta)(x - x_j^i) \\
&= \left(A_w(\delta)(x - x_1^2) + \cdots + A_w(\delta)(x - x_N^2) \right) dx \\
&= \sum_{i,j} \int_{x \in \mathbb{R}^n} A_w(\delta)(x - x_i^1) A_w(\delta)(x - x_j^2) dx \ . \qquad (24)
\end{aligned}$$

For the case of Gaussian κ as in (19),

$$\begin{aligned}
&\int_{x \in \mathbb{R}^n} A_w(\delta)(x - x^1) A_w(\delta)(x - x^2) dx \\
&= \int_{x \in \mathbb{R}^n} \exp(-(x - x^1)^T Q (x - x^1)) \\
&\qquad \cdot \exp(-(x - x^2)^T Q (x - x^2)) dx \\
&= \exp(-\frac{1}{2}(x^2 - x^1)^T Q (x^2 - x^1)) \\
&\quad \cdot \int_{x \in \mathbb{R}^n} \exp(-(x - \bar{x}^{12})^T Q (x - \bar{x}^{12}) dx \\
&= \exp(-\frac{1}{2}(x^2 - x^1)^T Q (x^2 - x^1)) \sqrt{\frac{\pi^n}{\det Q}} \qquad (25)
\end{aligned}$$

for any $x^1, x^2 \in \mathbb{R}^n$ (where $\bar{x}^{12} = (x^1 + x^2)/2$), and so

$$\begin{aligned}
&\kappa((x_1^1, \cdots, x_N^1), (x_1^2, \cdots, x_N^2)) \\
&= \sqrt{\frac{\pi^n}{\det Q}} \sum_{i,j} \exp(-\frac{1}{2}(x_j^2 - x_i^1)^T Q (x_j^2 - x_i^1)) \ . \qquad (26)
\end{aligned}$$

More generally, we may be interested in allowing indicator distributions of the form (6). In this case we have another embedding Φ_2 which maps (6), thought of as a *formal sum*, to the distribution encoded by the same, and its corresponding kernel

is

$$\kappa_{\Phi_2}(c_1^1\delta(x-\xi_1^1)+\cdots+c_K^1\delta(x-\xi_{K^1}^1),$$
$$c_1^2\delta(x-\xi_1^2)+\cdots+c_K^2\delta(x-\xi_{K^2}^2))$$
$$=\sqrt{\frac{\pi^n}{\det Q}}\sum_{i,j}c_i^1c_j^2\exp(-\frac{1}{2}(x_j^2-x_i^1)^TQ(x_j^2-x_i^1)) \tag{27}$$

which differs from (26) by the factors c_i^1, c_j^2 which are now included.

The significance of the kernel function we have obtained is that it gives the original indexed representation a permutation-invariant geometry which can be used to reason about multiagent control laws without necessarily needing to work at the level of partial differential equations. Moreover this geometry can be understood concretely in terms of indicator distributions, and this ties the Eulerian and Lagrangian approaches together.

3.1.4 A Finite-State-Space Analogue

One interpretation of the Eulerian view of multiagent systems presented so far is that, rather than thinking of agents as making decisions about which actions to take, one can instead view the states themselves as making decisions about how many agents should be entering or leaving them – subject to the dynamical constraints imposed by the number of agents in each state. We may take this interpretation very literally, and consider situations in which, e.g., rooms of a building decide which robots should enter them at any time, or in which sectors of a warzone command various autonomous support vehicles to enter or leave them in response to changing demands or in order to meet an objective.

In short, what we are considering is "dumb robots in a smart environment."

A qualitative observation that motivates this is the complementary behavior of the mass distribution m of the agents (the Eulerian setting) to that of their joint state vector x (the Lagrangian setting). We observe that in the Eulerian setting, consensus corresponds to a very "peaky" distribution, in which all mass is concentrated

at one point, whereas in the Lagrangian setting, consensus corresponds to a "flat" distribution of states over agents, in which $x_1 = x_2 = \cdots = x_N$. Hence we can expect, more generally, that each controller in the Lagrangian setting corresponds to a "dual" controller in the Eulerian setting, and that certain control objectives may be easier to achieve in one setting or in the other; this is the idea explored in Section 3.1.4.1.

The finite state spaces of these examples – e.g. rooms of a building – also motivate the construction of a finite-state-space analogue to the indicator-distribution representation we have discussed so far. To this end, we assume the existence of a set $R = \{1, \cdots, N\}$ of rooms, connected in an undirected graph $\mathcal{G}_p = (R, E_p \subset R \times R)$, in which edges indicate physical paths – e.g., hallways – by which agents can move between them. We likewise assume that the rooms can communicate via some network, represented as another graph $\mathcal{G}_c = (R, E_c \subset R \times R)$. Finally, we associate to each room $i \in R$ a number $m_i \in \mathbb{R}$ of agents currently in that room, and thereby define the vector $m = (m_1, \cdots, m_N) \in \mathbb{R}^N$ [The relaxation to allow for a real (rather than only natural) number of agents in each room can be viewed as a limiting case for a very large number of agents]. Assuming either a discrete timestep or Lebesgue sampling, the dynamics of the resulting system are summarized,

$$m[k+1] = m[k] + Du[k] \quad \forall k \in \mathbb{N} \tag{28}$$

subject to the elementwise state constraint $m[k] \geq 0$ for all $k \in \mathbb{N}$, where D is the incidence matrix associated with $\mathcal{G}_p$ and $u[k] \in \mathbb{R}^{|E_p|}$.

3.1.4.1 Example: Vacuuming an Office Building

As a particular concrete example, suppose that each room of an office building has been outfitted with a short-range (e.g., Bluetooth) wireless access point, and that we have a number of vacuum-cleaning robots that we would like the access points to deterministically direct throughout the building. The goal in this case, in order to minimize the amount of dirt left in the worst-cleaned room, is to achieve a uniform

distribution of robots over the rooms. The question then becomes how the rooms of the building, in a distributed way and while respecting state constraints, can choose to direct robots between themselves so that the robots are eventually distributed uniformly throughout the building. This is a version of the *coverage problem*, and it will turn out to be particularly easy to solve in the Eulerian setting.

For the purposes of this example we will assume $\mathcal{G}_p = \mathcal{G}_c = \mathcal{G}$ – i.e., that the physical and network topologies are the same – in which case the controller

$$u[k] = -\gamma D^T m[k] \tag{29}$$

for some $\gamma > 0$ gives the closed-loop dynamics

$$\begin{aligned} m[k+1] &= (I - \gamma D D^T) m[k] \\ &= (I - \gamma L) m[k] \end{aligned} \tag{30}$$

where L is the graph Laplacian for $\mathcal{G}$. Since L is positive semidefinite (see [72]), the eigenvalues of the closed-loop system matrix lie within the unit circle for sufficiently small γ (for all but the $\mathbf{1}$ eigenvector, whose eigenvalue is exactly 1), e.g. $\gamma = \frac{1}{2}||L||_2$, and so the system is stable. Moreover, it can be seen that the given controller satisfies the state constraint $m[k] > 0 \;\; \forall k \in \mathbb{N}$.

3.1.5 Index-Free Multiagent Systems via Mass Density Functions: Conclusions

We have stripped agent identities from the multiagent modeling machinery by employing an indicator function representation, and in so doing arrived at an integro-differential model for multiagent systems which parallels the now-standard graph-theoretic constructions. Along the way, we proved stability and conservation properties from within a continuum model, and, guided by our permutation-invariant representation and the so-called kernel trick, endowed the traditional vector state space with a permutation-invariant geometry. Finally, we illustrated a qualitative

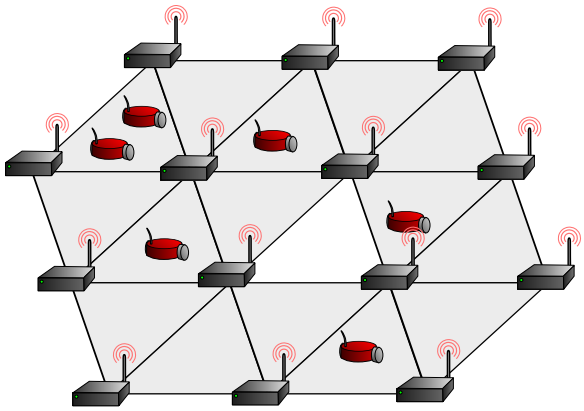

Figure 5: Multiple mobile robots (red) are directed throughout a triangulated environment with the help of wireless base stations (dark gray).

duality between the Eulerian and Lagrangian approaches by way of a finite-state-space analogue, which demonstrated that for certain problems a literal interpretation of the Eulerian approach can result in very simple controllers.

3.2 Distributed-Infrastructure Routing

Heartened by our initial success with simple, Laplacian-based controllers designed from the Eulerian perspective, we now aim to solve problems that involve not just convergence of agents to a static deployment in the environment, but interesting motions in the steady state. The purpose will be to route agents throughout an environment, while satisfying various safety and efficiency constraints. Figure 5 shows a "cartoon" of the envisioned scenario, in which static infrastructure nodes, like wireless base stations, communicate with one another in consensus-like protocols to synthesize controllers that are passed to mobile agents, that move within an environment represented as a simplicial complex.

3.2.1 One-Dimensional Models: Analogies

We will first consider "one-dimensional" models – those that represent the movement of agents throughout an environment by flows on *edges* of a graph. After developing algorithms to solve multiagent routing problems in this one-dimensional case, we will,

in subsequent chapters, address the two-dimensional case – in which controllers are constructed to move agents within triangular regions. Higher-dimensional analogues – in which tetrahedra, pentachora, etc, take the place of triangles – are then also possible.

In this and subsequent sections, we will be strongly motivated by close analogies between k-chains of different orders, and objects defined on differentiable manifolds.

A first set of analogies relates to the use of graphs as models for environments. Here, a vertex is the analogue of a point on a smooth manifold, and an edge is the analogue of an arclength-parametrized curve or unit tangent vector; here, *upper*-adjacency represents topology. A 0-chain is the analogue of a scalar field; its coefficients are values assigned to the corresponding vertices. A 1-chain is the analogue of a vector field; it can be thought of as assigning a directed flow to each edge. The coboundary operator $\delta_1^* : C_0(K) \to C_1(K)$ is the analogue of the gradient operator, and the boundary operator $\delta_1 : C_1(K) \to C_0(K)$ is the analogue of the divergence operator. Just as the Laplacian on $\mathbb{R}^n$ factors as $\nabla = \text{div grad}$, so too does the zeroeth combinatorial Laplacian factor into the analogous combinatorial operators, as $\mathcal{L}_0 = \delta_1^* \circ \delta_1$.

We will make a dual analogy for simplicial 2-complexes. Here, a triangle is the analogue of a point on a smooth manifold, and an edge or face is the analogue of a unit tangent vector; here, *lower*-adjacency represents topology. A 2-chain is the analogue of a scalar field. A 1-chain is the analogue of a vector field; it represents a directed flux across each face. The coboundary operator $\delta_1^* : C_0(K) \to C_1(K)$ is the analogue of the gradient operator, and the boundary operator $\delta_1 : C_1(K) \to C_0(K)$ is the analogue of the divergence operator.

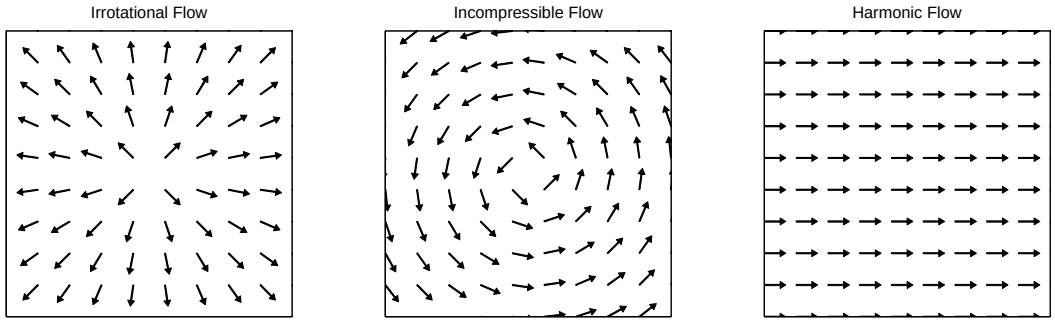

Figure 6: Prototypical irrotational (left), incompressible (center), and harmonic (right) vector fields on $\mathbb{R}^2$.

3.2.2 Helmholtz-Hodge Decomposition

The Helmholtz-Hodge decomposition of a vector field $v : \mathbb{R}^3 \to \mathbb{R}^3$ is its unique representation as the sum

$$v = v_c + v_r + v_h \tag{31}$$

with div $v_c \neq 0$, curl $v_c = 0$; div $v_r = 0$, curl $v_r \neq 0$; and div $v_h = 0$, curl $v_h = 0$. From a functional analysis perspective, the three terms are projections of v onto three orthogonal linear subspaces of the space of vector fields on $\mathbb{R}^3$. The three terms are the the *curl-free*, *divergence-free*, and *harmonic* components, respectively. The first represents sources and sinks, the second vortices, and the third global flows representing the topology of the space, and illustrated in Figure 6.

On a simplicial 1-complex (i.e., a graph) G, we can compute an analogous decomposition of a 1-chain $v \in C_1(G)$ as

$$v = v_{\mathrm{c}} + v_{\mathrm{r}} \tag{32}$$

with $v_{\mathrm{c}} \perp v_{\mathrm{r}}$ under the inner product (2); this is the subject of section 3.2.3. Note that by working with the 3-clique complex of a graph – a simplicial 2-complex – it is possible to further decompose v into a total of three components, including an analogue to the harmonic component of (31); this is the path taken in e.g. [50], but it comes at the cost of treating edges rather than nodes as the agents that perform computation.

3.2.3 Hodge Decomposition on Graphs

From Hilbert's Projection Lemma, we know that orthogonal projections are least-squares solutions to linear equations. In particular, the orthogonal projection of a 1-chain $v \in C_1(G)$ onto its curl-free component can be found from the least-squares solution to the equation,

$$\delta_1^*(G)p = v \ . \tag{33}$$

We use $p \in C_0(G)$ for the unknown variable because it corresponds to pressure in fluid dynamics. The solution is readily found to be,

$$p = (\delta_1(G)\delta_1^*(G))^\dagger \delta_1(G)v \tag{34}$$

$$= \mathcal{L}_0^\dagger(G)\delta_1 v \tag{35}$$

where $(\cdot)^\dagger$ denotes the pseudoinverse operation. [4] Once p is known, the curl-free component is reconstructed easily as

$$v_c = \delta_1^*(G)p \ . \tag{36}$$

This decomposition appears both in the swarm control algorithms to be presented in the subsequent sections, but also in preference learning (see e.g. [50]; preference learning is discussed in more detail in Chapter 5).

What is interesting is that consensus dynamics solve the equation (33), as described in the following theorem:

Theorem 3. *The forced Laplacian dynamics*

$$\dot{p} = -\mathcal{L}_0(G)p + \delta_1(G)v \tag{37}$$

converge asymptotically to the solution (35) of (33) if $p(0) = 0$.

[4] For the (matrix representation of the) graph Laplacian of a connected graph, this is the inverse restricted to $\text{span}\{\mathbf{1}\}^\perp$. I.e., $L^\dagger = (L + \frac{1}{n}\mathbf{1}\mathbf{1}^T)^{-1} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$.

Proof : The ODE (37) can be written as

$$\dot{p} = -\text{grad}_p \, \frac{1}{2}||\delta_1^*(G)p - v||^2 \tag{38}$$

which are precisely the gradient descent dynamics needed to solve (33) (Here, the norm is that induced by the inner product (2)). Since the quadratic form is convex on the quotient space $C_0(G)/\mathcal{N}(\mathcal{L}_0(G))$, gradient descent converges in that quotient space regardless of initial condition, and since $p(0) = 0$, the component of p in $\mathcal{N}(\mathcal{L}_0(G))$ remains zero for all time. ∎

The important message is that the familiar Laplacian dynamics, when forced, solve the normal equations, and give a spatially-distributed way to asymptotically compute p.

The divergence-free component of the 1-chain v, likewise, is the projection of v onto image$\{\delta_1^*(G)\}^\perp$. Hence it can be found as,

$$v_r = v - v_c = v - \delta_1^* p \tag{39}$$

from the same p.

Due to the important role of discrete Laplacian operators in this and subsequent sections, before proceeding we will provide an alternative, energy-function–based description of these operators, which will additionally enable us to define a number of novel Laplacian-like operators of our own.

3.2.4 Laplacian Operators and Energy Functions

An equivalent way to introduce the symmetric Laplacian operators is by means of particular scalar-valued functions, which we will refer to as *energy functions*. Before proceeding, it will also be useful to define the *boundary subcomplex* $B(K)$ of K, which consists of those faces that agents cannot cross:

Definition 3.2.1. *The* boundary subcomplex $B(K)$ *of a pure simplicial n-complex*

K is the $n-1$-subcomplex,

$$B(K) = \mathrm{cl}\left\{ \sigma \in \Sigma_{n-1}(K) \;\middle|\; \begin{array}{l} \sigma \;\; \textit{has fewer than} \\ \textit{two cofaces.} \end{array} \right\} \tag{40}$$

where cl *denotes simplicial closure.*

Note that the $(n-1)$-skeleton of K decomposes as a union of $B(K)$ with the $(n-1)$-skeleton of $F(K)$, which are disjoint with one another. Simply put, each $(n-1)$-simplex is either a face that can be crossed, or one that cannot be crossed.

With the boundary subcomplex defined, we are now in a position to define the *energy functions*:

Definition 3.2.2. *For an abstract simplicial complex K, the k-th* energy function *$\mathcal{E}_k(K) : C_k(K) \to \mathbb{R}$ is defined by,*

$$\begin{aligned} \mathcal{E}_k(K)(x) = \quad & \frac{1}{2} \sum_{\Delta \in \Sigma_{k+1}(K)} \langle \delta_{k+1}(\Delta), x \rangle^2 \\ & + \frac{1}{2} \sum_{\sigma \in \Sigma_{k-1}(K)} \langle \delta_k^*(\sigma), x \rangle^2 \;\; , \end{aligned} \tag{41}$$

with the convention that $\Sigma_i(K) = \emptyset$ for all $i < 0$, and that summations over the empty set evaluate to zero.

By way of these energy functions, the (standard) combinatorial Laplacian can then be defined simply:

Definition 3.2.3. *For an abstract simplicial complex K, the k-th* combinatorial Laplacian *$\mathcal{L}_k(K) : C_k(K) \to C_k(K)$ is the Hessian of the k-th energy function, $\mathcal{E}_k(K)$.*

It will also be useful to define generalizations of the energy functions that omit certain terms from the summations of Definition 3.2.2, as well as corresponding Laplacian operators. The first of these is nonstandard:

Definition 3.2.4. *Let K be an abstract simplicial complex and $L \subset K$ be a subcomplex of K. Then the k-th* restricted energy function $\mathcal{E}_k(K, L)$ *is defined by,*

$$\begin{aligned}\mathcal{E}_k(K,L)(x) = \quad & \frac{1}{2} \sum_{\Delta \in \Sigma_{k+1}(K)/\Sigma_{k+1}(L)} \langle \delta_{k+1}(\Delta), x \rangle^2 \\ & + \frac{1}{2} \sum_{\sigma \in \Sigma_{k-1}(K)/\Sigma_{k-1}(L)} \langle \delta_k^*(\sigma), x \rangle^2 \\ & + \frac{1}{2} \sum_{\sigma \in \Sigma_k(L)} \langle \sigma, x \rangle^2 \end{aligned} \tag{42}$$

The restricted Laplacian is then defined in much the same way as before:

Definition 3.2.5. *For an abstract simplicial complex K, and subcomplex $B \subset K$, the k-th* restricted combinatorial Laplacian $\mathcal{L}_k(K, B) : C_k(K) \to C_k(K)$ *is the Hessian of the k-th restricted energy function, $\mathcal{E}_k(K, B)$.*

Moreover, we will refer to the special case of $L(K, B(K))$ as the *boundary-restricted Laplacian* corresponding to K.

The boundary-restricted Laplacian is particularly useful because it guarantees zero flow out of the complex, while still characterizing the homology group of the complex, as described by the next theorem:

Theorem 4. *Let $x \in C_{k-1}(K)$ be a $(k-1)$-chain on a pure k-complex K. If x is zero on $B(K)$, then x is in the null space of the boundary-restricted Laplacian $\mathcal{L}_{k-1}(K, B(K))$ if and only if it is in the null space of the standard Laplacian $\mathcal{L}_{k-1}(K)$.*

Proof. Since x is zero on $B(K)$, all terms of the third summation of (42) are zero; consequently, we need only consider the first two summations. If $x \in \mathcal{N}\,\mathcal{L}(K)$, it then follows that $x \in \mathcal{N}\,\mathcal{L}(K, B(K))$, since the terms of the first two summations in (42) are a subset of those in (41), and each term is positive. To show the converse, we note that the only terms that appear in (41) but not in the first two summations of (42) correspond to faces in $B(K)$, and, since x is zero on $B(K)$, these terms are zero. ∎

Finally, we define a directed zeroeth Laplacian operator:

Definition 3.2.6. *Let $G = (V, E)$ be a directed graph. Then the* directed graph Laplacian $L(G) : C_0(V) \to C_0(V)$ *is defined by,*

$$[L(G)(x)]_i = \sum_{j|(v_j,v_i)\in E} (x_i - x_j) \ . \tag{43}$$

Note that, since G in Definition 3.2.6 is directed, it may be that $(v_i, v_j) \in E$ while $(v_j, v_i) \notin E$, in which case $L(G)$ is not symmetric.

With the necessary Laplacian operators thus defined, we now consider the computation of face fluxes that serve as representatives of the first homology group.

3.2.5 Two-Dimensional Models

We now shift our attention from one- to two- dimensional models of the environment; these described by simplicial 2-complexes. We will describe a method for generating incompressible vector fields in their Rips Shadows as Hamiltonian vector fields, and for computing a single global streamfunction that generates these.

In this line of thought, *agents are 2-simplexes*. For the case of air traffic control, this represents the idea that each simplex is a region of airspace under the authority of a particular controller on the ground, and that it is the job of these automated ground controllers to agree in a distributed way how airplanes should be routed among themselves.

We will assume that the graph G of the previous sections is the lower-adjacency graph of the triangles of a pure simplicial 2-complex – i.e., that, given a 2-complex K, $V(G) = \Sigma_2(K)$, and (Δ_1, Δ_2) is an edge of G if and only if $\Delta_1 \smile \Delta_2$ in K. Equivalently, G is the subgraph of the dual graph to the 1-skeleton of K obtained by deleting the "outside vertex" (v_0 in Figure 7).

In what follows, we will produce an incompressible flow over $\mathcal{R}(K)$ by computing a particular 0-chain over K. To do this, we first introduce a family of local flows

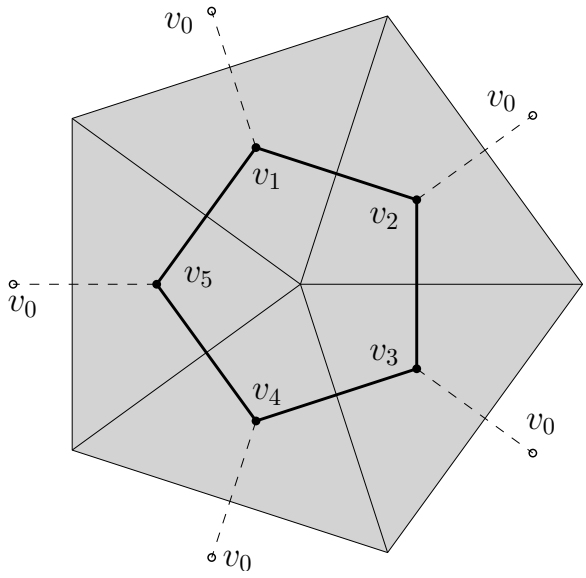

Figure 7: Given a planar simplicial 2-complex K (gray), G is the lower-adjacency graph (bold lines) of the triangles. It is a subgraph of the dual graph $\mathcal{G}$ (bold and dashed lines) to the 1-skeleton of K (thin solid lines), denoted $\mathcal{G}^*$. (Note that the five copies of v_0 (circles) are identified.)

defined on the individual k-simplices (this is the subject of Section 3.2.5.1), and then compute a global 0-chain over K (Section 3.2.5.3) representing a streamfunction.

3.2.5.1 Local vector fields

In this section we will describe the individual building blocks for our global vector field. In particular, given a 0-chain over the vertices of a simplex, we will produce an incompressible flow within the simplex. This is done by using barycentric interpolation to create a streamfunction over the simplex, and defining a Hamiltonian vector field along this streamfunction.

Let $x_1, x_2, x_3 \in \mathbb{R}^2$ be the vertices of a realization of an oriented 2-simplex $\Delta = [v_0, v_1, v_2]$, Defining $X = [x_1, x_2, x_3] \in \mathbb{R}^{2\times 3}$, the barycentric coordinates $b \in \mathbb{R}^3$ of a point $x \in \mathbb{R}^2$ are the unique solution to the equations,

$$Xb = x \tag{44}$$

$$\mathbf{1}^T b = 1 \ . \tag{45}$$

It is also convenient to define the inverse matrices $B_1 \in \mathbb{R}^{3\times 2}$ and $B_2 \in \mathbb{R}^{3\times 1}$ by[5]

$$\begin{bmatrix} X \\ \mathbf{1}^T \end{bmatrix}^{-1} = \begin{bmatrix} B_1 & B_2 \end{bmatrix} . \tag{46}$$

Then, letting $c_0v_0 + c_1v_1 + c_2v_2$ be a 0-chain on Δ and $c = (c_0, c_1, c_2) \in \mathbb{R}^3$, we define a scalar field $\phi(\Delta) : \mathbb{R}^2 \to \mathbb{R}$ over the Rips Shadow of Δ by

$$\phi(\Delta)(x) = c^T \left(B_1 x + B_2\right) . \tag{47}$$

We will call $\phi(\Delta)$ the *local streamfunction* corresponding to the simplex Δ.

Finally, the Hamiltonian dynamics corresponding to $\phi(\Delta)$ are defined, in Cartesian coordinates, as

$$\begin{aligned} \dot{x} &= J \operatorname{grad} \phi(\Delta) \\ &= JB_1^T c \end{aligned} \tag{48}$$

or in barycentric coordinates as,

$$\begin{aligned} \dot{b} &= B_1 J B_1^T c \\ &\triangleq A(\Delta) \end{aligned} \tag{49}$$

where $J \in \mathbb{R}^{2\times 2}$ is the matrix representation of the symplectic form $(a, b) \mapsto \det([a, b])$. I.e., $J = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$, so that, for all $a, b \in \mathbb{R}^2$, $a^T Jb = \det([a, b])$ (where $[a, b] \in \mathbb{R}^{2\times 2}$ denotes the matrix with a and b as columns).

Lemma 4. *The vector field (48) is divergence-free within each triangle.*

Proof : The vector field $x \mapsto JB_1^T c$ is constant in x, so its divergence is zero. ∎

We will now use these per-simplex building blocks to assemble a single global vector field on K.

[5]The inverse has a nice interpretation: b_i is the ratio of the volume of the simplex with x substituted for x_i, to that of the original simplex.

3.2.5.2 A global vector field

Under the assumption that the interiors of the Rips Shadows of all the simplices are disjoint, we define the piecewise vector field $\nu : \mathcal{R}(K) \to \mathbb{R}^3$ in barycentric coordinates by,

$$\nu(x) = \left\{ \; A(\Delta) \quad \text{if} \quad x \in \mathcal{R}(\Delta) \; \forall \Delta \in K \quad . \right. \tag{50}$$

In the section that follows, we will show that this vector field is *globally* divergence-free by demonstrating the existence of a single global streamfunction. Moreover, we will give a distributed algorithm to compute this streamfunction.

Before proceeding, however, we would like to point out that, already, (50) by itself constitutes a single hybrid controller for the vehicles: Each vehicle looks up which 2-simplex Δ it is in, requests the vector $A(\Delta)$ from Δ, and then follows that vector field.

3.2.5.3 The global stream function

We would like to construct a global streamfunction $\phi : \mathcal{R}(K) \to \mathbb{R}$ of the form,

$$\phi(x) = \left\{ \; \phi(\Delta)(x) \quad \text{if} \quad x \in \mathcal{R}(\Delta) \; \forall \Delta \in K \right. \tag{51}$$

that produces the vector field (50) – for some *global* 0-chain over K. In the following sections, we prove that such a 0-chain exists, and give algorithms for computing it.

Existence and Properties

Definition 3.2.7. *Given an oriented simplicial k-complex K, a vector field (in barycentric coordinates) $v : \mathcal{R}(K) \to \mathbb{R}^3$* agrees with *a $(k-1)$-chain v if, for each simplex $\Delta \in \Sigma_{k-1}(K)$, the flux of v across $\mathcal{R}(\Delta)$ equals $\langle v, \Delta \rangle$.*

Theorem 5. *If v is a divergence-free 1-chain over G, then there exists a 0-chain over K that induces a Hamiltonian vector field agreeing with v on the Rips Shadow of K.*

Proof : Since the edge flow v is in the cycle space of G and $G \subset \mathcal{G}$, it is in the cycle space of $\mathcal{G}$. Then, by cycle-cut duality, it is in the cut space of $\mathcal{G}^*$, the 1-skeleton of K. Consequently there exists a vector c' in the vertex space of G^*, or equivalently a 0-chain c over K, whose coboundary is v. ∎

3.2.5.3.1 Distributed computation of a global stream function

Method 1 This first method serves to motivate the second. As in section 3.2.3, we are faced with the problem of computing a 0-chain whose boundary best approximates a given 1-chain; hence the global 0-chain $c \in C_0(K)$ can be computed using the gradient descent dynamics,

$$\dot{c} = -\mathcal{L}_0(K)c + \delta_1(K)v \tag{52}$$

where now c is a 0-chain over the vertices of K rather than of G, and the operators $\mathcal{L}_0, \delta_1$ likewise correspond to K. An issue with this approach is that vertices of K are shared by multiple agents – triangles – so an additional synchronization protocol is required for an actual implementation. The next method avoids this messiness, and is much more compatible with the reality that it is triangles, not vertices, that represent agents.

Method 2 Within a *single* oriented 2-simplex Δ, the problem of computing 0-chains with given boundaries is straightforward. Let $c \in C_0(\Delta)$ and $v \in C_1(\Delta)$ be 0- and 1-chains over Δ representing streamfunction values and face fluxes, respectively. The problem is that of solving the equation

$$\delta_1^*(\Delta)c = v, \tag{53}$$

where $\delta_1(\Delta)$ has the matrix representation

$$E_3 \triangleq \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} . \tag{54}$$

Since the matrix E_3^T has a 1-dimensional null space spanned by $\mathbf{1}$, there is a family of solutions,

$$c = [\delta_1^*(\Delta)]^\dagger v + \mathbf{1}s \tag{55}$$

where $\mathbf{1} \in C_0(\Delta)$ is the 0-chain that assigns a 1 to each vertex.[6] What this means is that, if a single agent – a triangle – knows its face fluxes, then it can independently determine what the 0-chain over its vertices should be, up to a constant. The coordination problem then is only to determine that scalar s for each triangle – i.e., a 2-chain over K, or, equivalently, a 0-chain over G.

What need the values $s_1, \cdots, s_N$ of the different triangles satisfy? Namely, for two consistently-oriented simplices indexed i and j, sharing a face that is the k^{th} face of simplex i and the l^{th} face of simplex j,

$$s_i - s_j = -\frac{1}{6}\left[D_k(\bar{v}_i) - D_l(\bar{v}_j)\right] \triangleq w_{ij} \tag{56}$$

where $\bar{v}_j \in \mathbb{R}^3$ is the vector representation of the restriction of the 1-chain v to the simplex j, and $D_k(\bar{v})$ is defined by,

$$[D_0(\bar{v}), D_1(\bar{v}), D_2(\bar{v})]^T = E_3[\bar{v}_0, \bar{v}_1, \bar{v}_2]^T . \tag{57}$$

The skew-symmetric matrix $W = [w_{ij}]_{ij}$ itself encodes a 1-chain over $\mathcal{G}$. The problem has thus been reduced to computing a 0-chain $s \in C_0(G)$ – that with coefficients $s_1, \cdots, s_N$ – given a 1-chain, $w \in C_1(G)$ – whose coefficients come from W – that is

[6] Note that the matrix representation of the pseudoinverse in (55) is particularly simple: $(E_3^T)^\dagger = \frac{1}{3}E_3$.

to be its boundary. Hence, s can be computed asymptotically by the system,

$$\dot{s} = -\mathcal{L}_0(G)s + \delta_1(G)w \tag{58}$$

much as before.

3.2.6 A Combined Algorithm

The two distributed computations described in the previous sections can be performed simultaneously within the network, and stability properties are maintained. This is the subject of the following theorem.

Theorem 6. *The solution of the ODE*

$$\begin{bmatrix} \dot{s} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} -\mathcal{L}_0 & -\delta_1 \mathcal{D} \delta_1^* \\ 0 & -\mathcal{L}_0 \end{bmatrix} \begin{bmatrix} s \\ p \end{bmatrix} + \begin{bmatrix} \delta_1 \mathcal{D} \\ \delta_1 \end{bmatrix} v \tag{59}$$

(where $\mathcal{D}$ is the linear operator that produces the 1-chain w following (56)), converges asymptotically to a vector in $C_0(G) \times C_0(G)$ that solves the equations (35) and (56).

Proof : The system matrix in (59), which we will refer to as A, is block-upper-triangular, so its eigenvalues are those of its diagonal blocks. Those in turn are graph Laplacians, which are known to be positive semi-definite (see e.g. [72]). Consequently, (59) converges asymptotically to a solution (s, p) provided it has no Jordan blocks larger than 1×1 – a possibility that is ruled out since $\text{image}(\delta_1 \mathcal{D} \delta_1^*) \perp \mathcal{N}(\mathcal{L}_0)$. ∎

3.2.7 Example: Air Traffic Control via Incompressible Flows

In this section, we explore the use of the the vector fields obtained in the previous sections to direct air traffic throughout an environment, in order to give a flavor for how the preceding theory can be applied. The idea will be to project control inputs onto the divergence-free subspace, using consensus dynamics as in Section 3.2.3; this ensures that aircraft don't "pile up" anywhere.

The aircraft are assumed to inhabit the nodes of the graph G (which correspond to different regions of airspace), and the edges encode which regions are adjacent. Let $M \in C_0(G)$ be a constant scalar field on G representing the capacity of each vertex – i.e., the number of aircraft that can safely share that airspace – and let $m : \mathbb{R}_+ \to C_0(G)$ be a time-varying 0-chain on G representing the number of aircraft at each vertex. Assuming $m(0) = M$ – i.e., that the airspace is initially filled to capacity – we investigate the general problem of directing the aircraft between them while maintaining the safety constraint $m(t) \leq M \ \forall t$.

For this example, we restrict our attention to the case when safety is ensured by maintaining $m(t) = M \ \forall t$ with equality. This is guaranteed by maintaining $\dot{m}(t) = 0 \ \forall t$, which in turn is satisfied by ensuring that the 1-chain describing the air traffic is divergence free. Two such problems naturally arise, from different projection operations, described in the following sections.

3.2.7.0.2 Least-squares approximation

Suppose an operator wishes to command the air traffic system with a particular reference vector field. One way in which the system can respond to this command is by providing the vector field that approximates the commanded field optimally in a least-squares sense while satisfying the incompressibility constraint. This is precisely the projection problem of Section 3.2.3, so the problem can be directly solved in a distributed fashion by the algorithm (37). We should note that, in order to do this, the operator need only communicate with two nodes per nonzero commanded edge flow; this is encoded by the product $\delta_1(G)v$.

3.2.7.0.3 Smallest divergence-free flow containing a particular component

A second way in which an operator's commanded vector fields may be used is by finding the smallest (in an l^2 sense) divergence-free flow containing the commanded flow $\bar{v}$ as a component; this is the lowest-energy safe holding pattern that guarantees

a certain amount of traffic on specified edges. In this case, we seek a solution to the constrained optimization problem,

$$\arg\min_{v \in C_1(G)} \frac{1}{2}||v||^2 \tag{60}$$

$$\text{s.t.} \quad \delta_1(G)v = 0 \qquad \text{Divergence-free} \tag{61}$$

$$\langle \bar{v}, v \rangle = ||\bar{v}||^2 \qquad \text{Contains component.} \tag{62}$$

Theorem 7. *Let P_r denote the l^2 projection operator for the divergence-free subspace of $C_1(G)$, which is computed by (37). Then for all $\bar{v} \neq 0$, the problem (60-62) either has the solution*

$$\frac{||\bar{v}||^2}{\langle P_r \bar{v}, \bar{v} \rangle} P_r \bar{v} \tag{63}$$

or is infeasible.

Proof: If $\bar{v} \perp \ker \delta_1(G)$, then (62) requires $v \notin \ker \delta_1(G)$. This contradicts (61), so in this case the problem is infeasible. Hence, without loss of generality, suppose $\bar{v} \notin (\ker \delta_1(G))^\perp$.

Any vector $v \in C_1(G)$ can be decomposed uniquely as $v = v_c + v_r$, with $v_c \in (\ker \delta_1(G))^\perp$ and $v_r \in \ker \delta_1(G)$. Furthermore, v_c can be uniquely decomposed as $v_c = v_{c,||} + v_{c,\perp}$, with $v_{c,||} \in \operatorname{span} P_r \bar{v}$ and $v_{c,\perp} \in (\operatorname{span} P_r \bar{v})^\perp$; hence a unique decomposition $v_c = v_{c,||} + v_{c,\perp} + v_r$ exists, and by the Pythagorean Theorem, $||v_c||^2 = ||v_{c,||}||^2 + ||v_{c,\perp}||^2 + ||v_r||^2$. By (61), $||v_r||^2 = 0$, and by (62), $||v_{c,||}||^2 = ||\bar{v}||^2$. Only $||v_{c,\perp}||^2$ remains free; the quantity $||v_c||^2$ is minimized when $||v_{c,\perp}||^2 = 0$. To summarize, we know that $v = v_{c,||}$ with $v_{c,||} \in \operatorname{span} P_r \bar{v}$, and that $||v_{c,||}||^2 = ||\bar{v}||^2$. Only one element of the vector space $C_1(G)$ satisfies these properties, and that is (63). ∎

3.2.8 Incompressible Flows: Numerical Example

To demonstrate the character of the results obtained with these methods, starting from a simplicial 2-complex K with second lower-adjacency graph G, we computed

the divergence-free projection of a commanded 1-chain on G with three nonzero elements, and the corresponding 0-chain on K and streamfunction on the Rips Shadow of K; this is shown in Figure 8. Note that the large commanded flow across a single face at the upper right of the complex is propagated through the "jughandle" at the upper right, and that the commanded flows lower in the complex in less confined areas result in pairs of vortices that have mostly local effects; nevertheless, small flows are produced throughout the complex. These qualitative characteristics are typical of the kinds of flows obtained: Where necessary, flows propagate globally, but otherwise most effects of a command are manifested locally. It is the pressure field that propagates this information; essentially, "shocks" are created across the faces where large flows are commanded, and elsewhere the pressure is smoothed across the complex by diffusion. The nonzero commanded flow at the upper right demonstrates this well; it creates a "shock" in the pressure field (black triangle next to white triangle), which diffusion spreads into linearly-decreasing pressure around the upper right "jughandle." Where vortices are produced, the streamfunction exhibits a pair of local extrema – a maximum for a clockwise vortex and a minimum for a counterclockwise one – as can be observed in the left part of the complex. Vehicles then follow level sets of the streamfunction around the environment.

3.2.9 Incompressible Flows: Implementation

A demonstration was produced using a collection of ten Khepera III mobile robots, in an indoor environment, as shown in Figure 9. Blue edges represent the boundary of the complex; internal edges are shown in purple. The input flows are generated by a human using a motion capture wand; whenever the wand's projection onto the floor crosses an internal edge, an input flow is added in the same direction across that edge, with magnitude an increasing function of the wand's speed. Video frames, with streamfunction and flows, are shown in Figure 10.

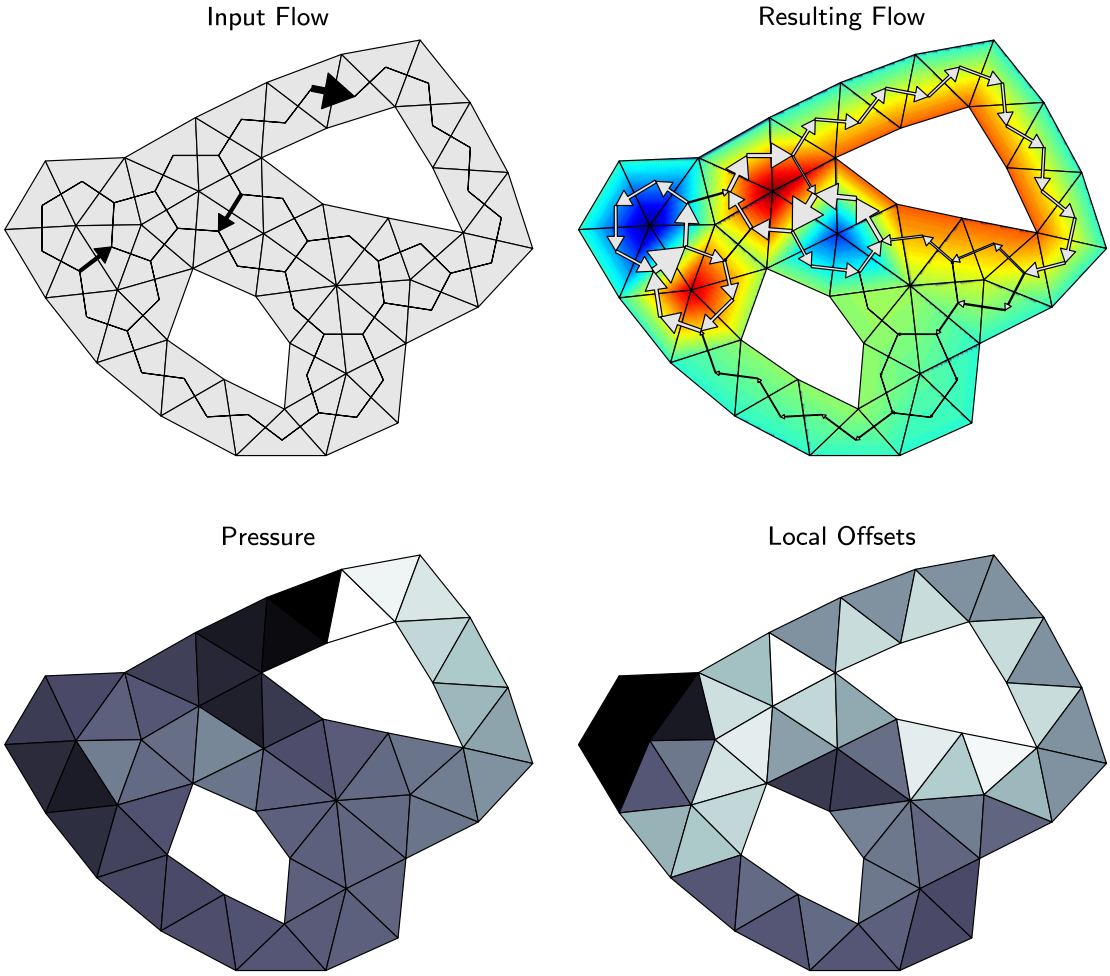

Figure 8: Computational results are shown. Given a flow as input (first plot; arrow sizes indicate flow magnitudes) on $\mathcal{G}$, a circulant flow on $\mathcal{G}$ and a streamfunction on the Rips Shadow of K are produced (second plot). The Lagrange multipliers for the cycle-space projection (third plot) are a close analogue of pressure in the dynamics of incompressible fluids. The streamfunction is computed locally at each triangle, requiring only the addition of a local offset (fourth plot), which is computed in a distributed fashion.

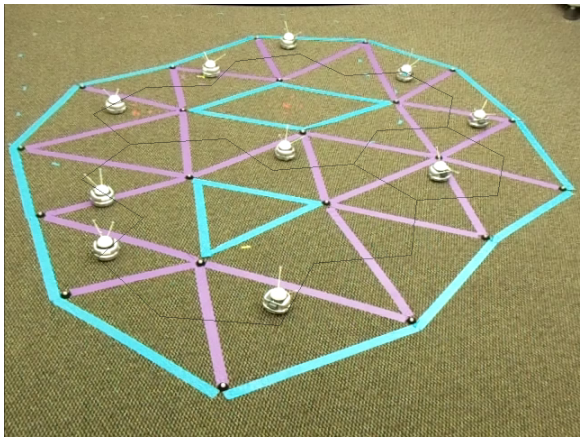

Figure 9: Incompressible-flow human-swarm interaction demo: The scenario.

3.2.10 Incompressible Flows: Summary

Given specified input flows, distributed consensus-like algorithms were described that compute divergence-free approximations. Then, these discrete flows were "lifted" to two-dimensional streamfunctions that generate vector fields over the entire Rips Shadows of corresponding simplicial 2-complexes. These flows mimic the behavior of incompressible fluids, and, since vehicles following them will never concentrate in any region, provide a useful method for coordinating collision-free navigation among large numbers of agents.

3.2.11 Harmonic Flows: or, Homological Patrol Strategies

We will now consider more complicated flow constraints. The goal is, as before, for the static agents $\Sigma_n(K)$, in a distributed fashion, to produce a family of continuous vector fields $\{v_i : \mathcal{R}(K,r) \to \mathbb{R}^n\}_{i=1}^M$ having nice properties, by which the mobile agents can circulate throughout the environment, as required for robot patrol or aircraft holding-pattern applications, but we will now require that the flows satisfy additional conditions.

In particular, we look for any vector field f in this family to satisfy three properties:

P1.1 The first of these, *uniform coverage*, insists that if the initial probability density

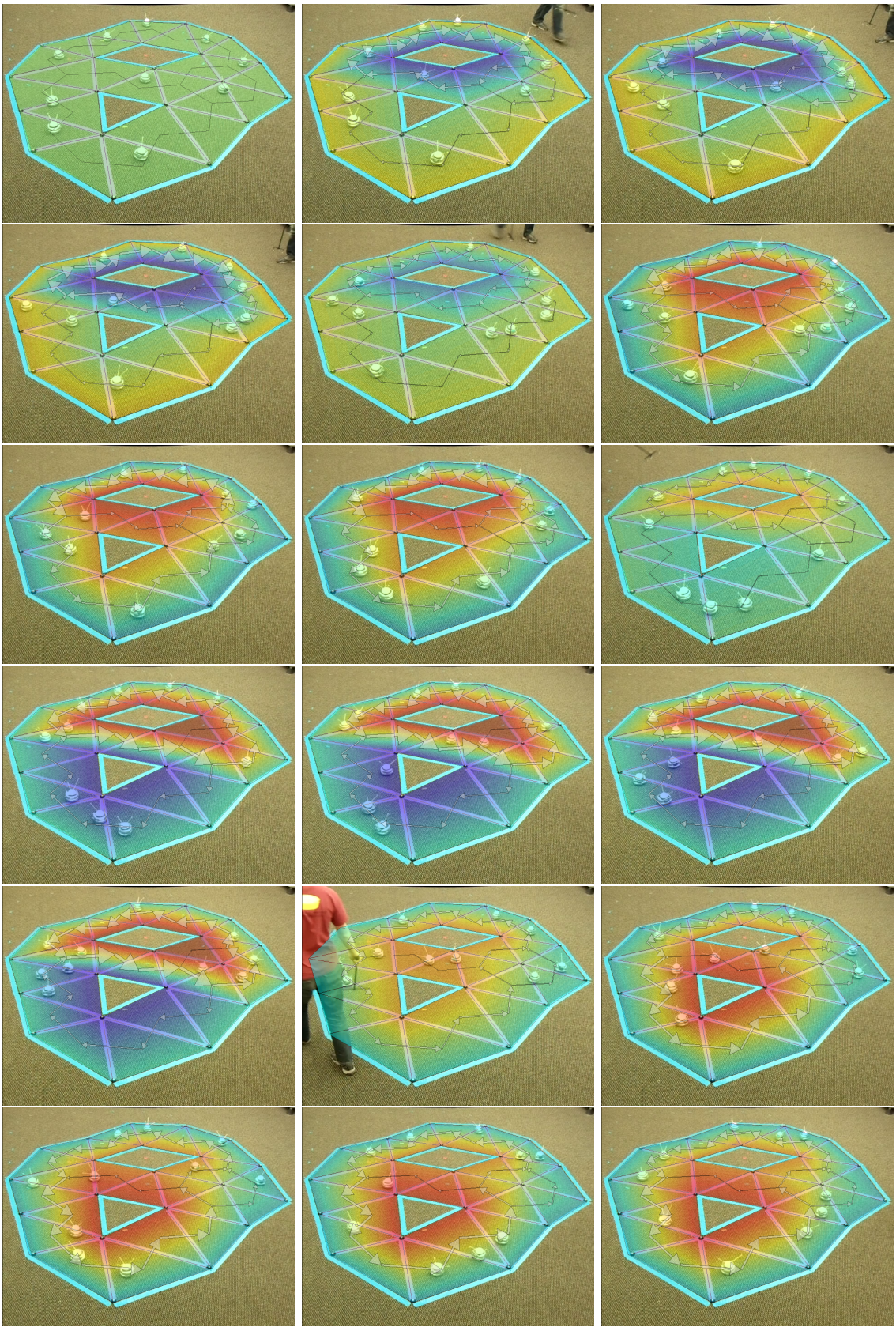

Figure 10: Incompressible-flow human-swarm interaction demo: Video frames, overlaid with flows (gray arrows) and streamfunctions (colored gradients; blues are low values and reds are high).

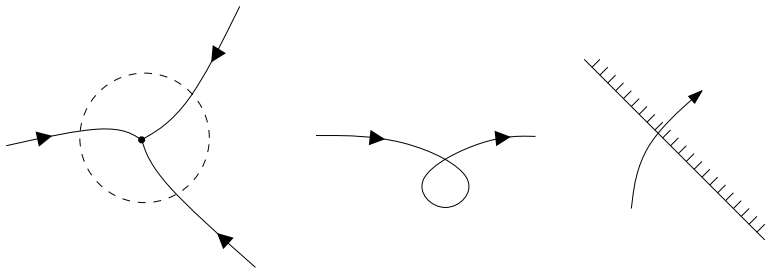

Figure 11: The vector fields produced avoid the concentration of agents within any one control volume (dashed circle) (left), paths with local loops (center), and collisions with the boundary of the complex (right).

of the agents is the uniform distribution $m(x, 0) = 1/\operatorname{Area}(\mathcal{R}(K))$ over the environment, that this condition persist; i.e., that $m(x, t) = 1/\operatorname{Area}\mathcal{R}(K)$ for all positions x and positive times t.

P1.2 The second, *no local cycles*, encourages that efficient paths without unnecessary loops be traced out by the agents following v, and is expressed by the requirement that no closed integral curve of f be contractible in $\mathcal{R}(K)$ to a point.

P1.3 The third, *zero boundary flux*, requires that agents not leave the environment. Formally, this means that f must have no component orthogonal to the *boundary* of the complex.

The requirements are illustrated by Figure 11.

This problem will be addressed in three parts. First, it is shown how existing distributed protocols can be used to produce face fluxes satisfying a *discrete* version of most (but not all) of the requirements P1. Next, symplectic vector fields are generated over the *continuous* space that are consistent with given flows, also in a distributed fashion. Finally, a simple, unified algorithm is presented that solves the continuous and discrete parts of the problem simultaneously, while also satisfying the remaining requirements of P1.

3.2.12 Distributed Computation of Homological Streamfunctions

In the next subsections, we will describe two classes of distributed methods for computing vector fields satisfying P1. The first, which serves to motivate the second, employs a method described in [75] to compute face fluxes within K, and then, as a separate step, adapts another distributed algorithm, described in [55], to compute vector fields that are consistent with those fluxes, via streamfunctions. The second, which is one of the contributions of this chapter, consists of considerably simplified, unified algorithms that compute the discrete flows and continuous controllers simultaneously, and that also satisfy additional requirements neglected by the first method.

We will make a departure from the previous section, in which 1-chains represented flows on edges, and a dual graph was constructed, by using $n-1$-chains on K (i.e., elements of $C_{n-1}(K)$) to, directly and without constructing a dual graph, represent flows of mobile agents across faces. The idea is that, if a simplex $\sigma \in \Sigma_{n-1}(K)$ appears in a chain with positive coefficient $c \in \mathbb{R}$, then agents are flowing across σ *from* the coface of σ with dissimilar orientation, *to* the coface with similar orientation.

In the subsequent sections, we will be interested in computing face fluxes in $C_{n-1}(F(K))$ that serve as representatives of the homology group $H_{n-1}(F(K))$. Later in this section, continuous control laws will be produced that achieve these fluxes.

3.2.12.1 Projection onto the first Homology Group

In this subsection, we describe a simple distributed algorithm for generating random elements of the homology group $H_k(K)$ without global knowledge of the graph topology.

Recalling that $H_k(K) = Z_k(K)/B_k(K)$, what we will do is produce unique representatives of elements of $H_k(K)$, that have a component in $Z_k(K)$ but not in $B_k(K)$. We are able to do this because a natural isomorphism exists between $H_k(K)$ and

the null space $\mathcal{N}\mathcal{L}_k(K)$ of the k-th combinatorial Laplacian. Since $\mathcal{N}\mathcal{L}_k(K) = Z_k(K) \cap B_k(K)^\perp$, an element $v \in \mathcal{N}\mathcal{L}_k(K)$ is the unique representative of an equivalence class in $H_k(K) = Z_k(K)/B_k(K)$ whose component in $B_k(K)$ is zero.

Since the restricted energy function $\mathcal{E}_1(K, B(K))$ is convex, gradient descent from any point converges asymptotically to $\mathcal{N}\mathcal{L}(K, B(K))_k$. Indeed, since $\mathcal{E}_1(K, B(K))$ has a quadratic but not a linear term, those gradient dynamics are simply,

$$\dot{x} = -\mathcal{L}_k(K, B(K))x \tag{64}$$

which, so long as k is not too large, constitute a distributed method for asymptotically computing the projection of a given k-chain $x(0)$ onto $\mathcal{N}\mathcal{L}_k = H_k(K)$. To elaborate, the sparsity pattern of $\mathcal{L}_k$ implies that this process requires $(k+1)$-hop communication in each round.

In [75], this property of the dynamics (64) was used to project a random 1-chain onto the homology group $H_1(K)$ in order to determine, with unit probability, whether it is trivial (i.e., has dimension zero). In this way, a sensor network could determine whether it contained any holes.

For our purposes, what matters is that this is an algorithm for producing unique representatives of elements of $H_1(K)$. With such a method thus in hand, we now turn our attention to the generation of continuous control laws from these edge flows. When the environment is two-dimensional (i.e., $n = 2$, and K is a pure 2-complex), these can be produced via *streamfunctions*, which we describe in the next subsection.

3.2.12.2 Hybrid Streamfunctions

We will compute streamfunctions over $\mathcal{R}(K, r)$ that induce vector fields that agree with the flows computed in the previous section in the following sense:

Definition 3.2.8. *Given an oriented simplicial k-complex K with realization $r : V(K) \to \mathbb{R}^n$, a vector field $v : \mathcal{R}(K) \to \mathbb{R}^n$* agrees with *a $(k-1)$-chain ν if, for each simplex $\Delta \in \Sigma_{k-1}(K)$, the flux of v across $\mathcal{R}(\Delta)$ equals $\langle \nu, \Delta \rangle$.*

Now that we have both (1) a method for computing 1-chains in $H_1(K)$ and (2) a method for computing streamfunctions that agree with them, in principle the two algorithms can simply be composed to produce homological streamfunctions satisfying P1.1 and P1.2. However, it is possible to perform a similar computation in a more unified manner, while additionally satisfying P1.3, as described in the next section.

3.2.12.3 Simple, unified algorithms

The key idea of this section is that the two algorithms just described can be unified by considering as a whole the properties that the 0-chain defining the streamfunction must satisfy; in the process, we will enforce the additional constraints imposed by P1.3. We will first consider a undirected, 2-hop algorithm that follows directly from our definitions, before introducing a directed, 1-hop algorithm that, remarkably, converges to the same set.

3.2.12.3.1 An undirected, 2-hop algorithm

The most straightforward approach, which results in an at-most-2–hop algorithm, results from composing the boundary and energy operators, as follows:

Theorem 8. *The dynamics,*

$$\dot{c}(t) = \delta_1(K)\mathcal{L}_1(K, B(K))\delta_1^*(K)c(t) \ \forall t > 0 \tag{65}$$

converge asymptotically from any initial condition $c(0) \in C_0(K)$ *to a value* $c(\infty) \in C_0(K)$ *such that, if* $x = \delta_1^*(K)c$*, then* $x \in \mathcal{N}\mathcal{L}(K, B(K))$*.*

Proof: The restricted energy function $\mathcal{E}(K, B(K))$ can be written,

$$\mathcal{E}(K, B(K))(x) = \frac{1}{2}x^*\mathcal{L}_1(K, B(K))x \ ; \tag{66}$$

consequently

$$\mathcal{E}(K, B(K))(\delta_1^*(K)c) = \frac{1}{2}c^*(t)\delta_1(K)\mathcal{L}_1(K, B(K))\delta_1^*(K)c(t) \tag{67}$$

whose gradient-descent dynamics with respect to c are (65). ∎

What this means is that, by running the simple, linear, 2-hop protocol (65), we can asymptotically compute a 0-chain, that induces a streamfunction, that induces a vector field satisfying P1. However, in the next section, we will see that this is also achievable with a directed, 1-hop algorithm.

3.2.12.3.2 A directed, 1-hop algorithm

Ultimately, we will produce a 0-chain satisfying the desired properties (i.e., that induces a vector field satisfying P1) by running Laplacian dynamics on a *directed* graph built from the complex K, which allows information to flow bidirectionally within the interior of the complex, as well as bidirectionally within the boundary, but only in a single direction between the two. This directed graph, which we will refer to as the *insulated 1-skeleton*, $G(K)$, only allows a 1-way flow of information from the boundary to the interior of the complex.

Definition 3.2.9. *The* insulated 1-skeleton $G(K) = (V, E)$ *of a pure simplicial n-complex K is the graph with vertex set $V = \Sigma_0(K)$, in which*

P2.1 *For all $a, b \in \Sigma_0(K/B(K))$, we have $(a, b), (b, a) \in E$ if and only if $(a, b) \in \Sigma_1(K)$ or $(b, a) \in \Sigma_1(K)$.*

P2.2 *For all $a, b \in \Sigma_0(B(K))$, we have $(a, b), (b, a) \in E$ if and only if $(a, b) \in \Sigma_1(B(K))$ or $(b, a) \in \Sigma_1(B(K))$*

P2.3 *For all $a \in \Sigma_0(B(K)), b \in \Sigma_0(K/B(K))$, we have $(a, b) \in E$ if and only if $(a, b) \in \Sigma_1(K)$ or $(b, a) \in \Sigma_1(K)$.*

P2.4 *For all $a \in \Sigma_0(K/B(K)), b \in \Sigma_0(B(K))$, we have $(a, b) \notin E$.*

Definition 3.2.9 is illustrated by Figure 3.2.12.3. In essence, one undirected graph links the vertices in the interior of the complex; another links those in its boundary;

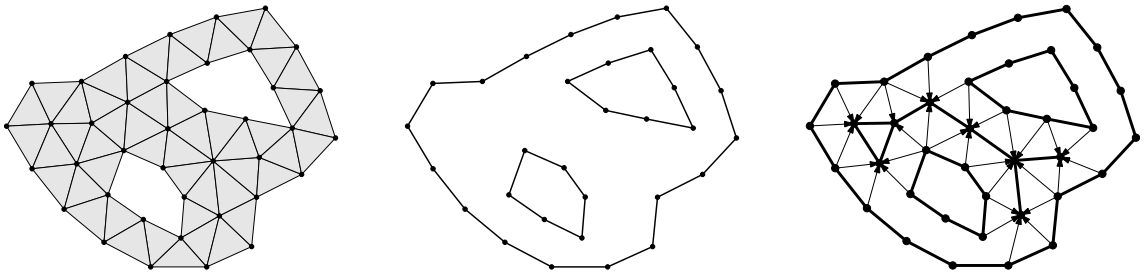

Figure 12: A pure simplicial 2-complex K (left), its *boundary subcomplex* $B(K)$ (center), and the corresponding *insulated 1-skeleton* $G(K)$ (right)

and edges between the two are directed from the boundary to the interior.

The insulated 1-skeleton $G(K)$ produced through definition 3.2.9 is illustrated by Figure 3.2.12.3. The next theorem explains how consensus dynamics on $G(K)$ can then be used to compute representatives of the homology group and streamfunctions together in a unified way.

Theorem 9. *Let K be a pure simplicial n-complex, $G(K)$ its insulated 1-skeleton, and L the directed Laplacian corresponding to $G(K)$. For any 0-chain $c_0 \in C_0(G(K))$, the directed Laplacian dynamics,*

$$\begin{aligned} \dot{c}(t) &= -Lc(t) \qquad \forall t > 0 \\ c(0) &= c_0 \end{aligned} \tag{68}$$

converge asymptotically to a 0-chain c_∞ such that $\delta_1^*(c_\infty) \in \mathcal{N}\,\mathcal{L}_1(K, B(K))$.

Proof : We must demonstrate that the system is stable, and that $\{c \mid \delta_1^* c \in \mathcal{N}\,\mathcal{L}_1(K, B(K))\}$ is its equilibrium set. First, we address stability. Without loss of generality, the dynamics (68) can be block-decomposed as,

$$\begin{bmatrix} \dot{x} \\ \dot{b} \end{bmatrix} = \begin{bmatrix} -(L_x + D_{xb}) & C \\ \mathbf{0} & -L_b \end{bmatrix} \begin{bmatrix} x \\ b \end{bmatrix} \tag{69}$$

where L_x is the (undirected) Laplacian for the undirected graph $G(K)/B(K)$, D_{xb} is a diagonal, positive-semidefinite matrix (representing the extra degree due to edges

connecting vertices in $G(K)/B(K)$ to $B(K)$), L_b is the (undirected) Laplacian for B, and C_{xb} is a coupling matrix representing edges from $G(K)/B(K)$ to $B(K)$.

To demonstrate stability, we must show that none of the Jordan blocks for the system's zero eigenvalues are larger than 1x1, and that all eigenvalues are nonpositive. First, note that, although the lower diagonal block, $-L_b$, does have a nontrivial nullspace, it, by the Spectral Theorem, is diagonalizable, so *none* of its Jordan blocks are larger than 1x1. Next, consider the upper diagonal block $-(L_x + D_{xb})$: Since there is at least one edge from B to each connected component of $G(K)/B(K)$, no zero-chain in the null space of L_x (i.e., that is constant on each connected component of $G(K)/B(K)$) can also be in the null space of D_{xb}, and consequently $-(L_x + D_{xb})$ is negative *definite*. Since the upper diagonal block has strictly negative eigenvalues, and, although the lower block does have zero eigenvalues, they correspond to Jordan blocks of size 1x1, the system is stable.

Next, we demonstrate that the equilibrium set is precisely the subspace $\{c \mid \delta_1^* c \in \mathcal{N}\mathcal{L}_1(K, B(K))\}$, by considering in turn each of the requirements that a vector in $\mathcal{N}\mathcal{L}_1(K, B(K))$ must satisfy, with reference to (42):

- *No circulation:* The first sum of (42) is zero if and only if, for each simplex $\Delta \in \Sigma_2(K)/\Sigma_2(B(K))$, we have $\langle \delta_2(\Delta), \delta_1^*(c_\infty)\rangle = 0$, or equivalently (by definition of an adjoint operator), if $\langle \delta_1\delta_2(\Delta), c_\infty\rangle = 0$. Since $\delta_k\delta_{k+1} \equiv 0$ for all $k \in \mathbb{N}$, this is automatically true.

- *No divergence in interior:* The second sum of (42) is zero if and only if, for each $\sigma \in \Sigma_0(K)/\Sigma_2(B(K))$, we have $\langle \delta_1^*(\sigma), \delta_1^*(c_\infty)\rangle = 0$, or, equivalently, if $\langle \sigma, \delta_1\delta_1^* c_\infty\rangle = \langle \sigma, \mathcal{L}_0(c_\infty)\rangle = 0$. This is precisely the condition, specified by the upper diagonal block of the system matrix, for vertices in $\sigma \in \Sigma_0(K)/\Sigma_2(B(K))$ to be at equilibrium.

- *Boundary condition:* The third sum of (42) is zero if and only if, for each

Input: Oriented simplicial complex K; $q \in \mathbb{N}$
Output: $c_{\infty,1}, \cdots, c_{\infty,q}$
Algorithm:

- For $i = 1$ to q
  - Generate random 0-chain $c_0 \in C_0(K)$, according to any probability distribution p on $C_0(K)$, the span of whose support is $C_0(K)$.
  - Run dynamics (68) or (65) from initial condition c_0 until convergence; store result as $c_{\infty,i}$.

Figure 13: Distributed algorithm for computing homological patrol strategies.

$\sigma \in \Sigma_1(B(K))$, we have $\langle \sigma, \delta_1^*(c_\infty) \rangle = 0$, or, equivalently, iff c_∞ is constant on each connected component of $B(K)$. The lower diagonal block of the system matrix is the standard zeroeth combinatorial Laplacian for $B(K)$, and its null space is exactly such zero-chains.

With stability guaranteed and the equilibrium set characterized, we conclude the proof. ∎

3.2.13 Infrastructure-Assisted Behavior Generation

In a two-dimensional environment with h holes,[7] the distributed homological-streamfunction generation algorithm of the previous section can be employed in a straightforward way to generate, with probability 1, an h-dimensional vector space of patrol behaviors. With the "heavy lifting" done by the distributed projection algorithm, the remainder of the behavior generation algorithm is exceedingly simple: So long as $q \geq h$, the resulting family of vector fields induced by the zero-chains $c_{\infty,1}), \cdots, c_{\infty,q}$ will span the space of vector fields satisfying P1.

A possible objection can be raised, which is that, if q is chosen according to a conservative upper bound on h, then the set of behaviors obtained will not be

[7] i.e., in a 2-complex whose first Betti number is h

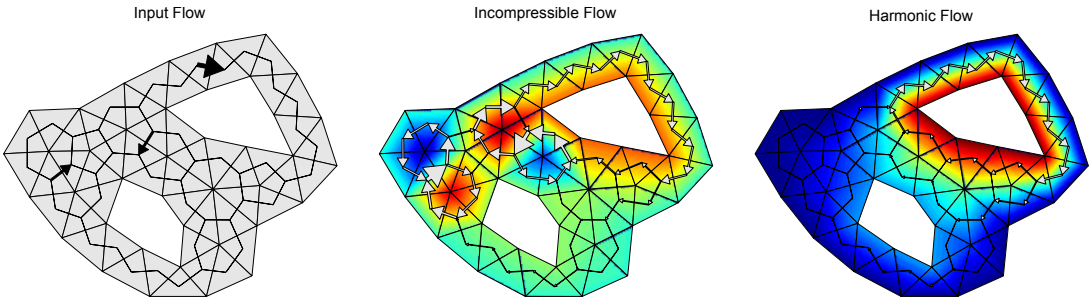

Figure 14: Depicted are a specified flow (left), and its projections onto the incompressible (center) and harmonic (right) subspaces. The harmonic flow described in this section (right) differs from the incompressible flow described in Section 3.2.8 (center), in that the former avoids the local vortices visible in the latter. In both cases, it is a collection of hybrid, piecewise-linear controllers that realize the flows. These controllers are produced as the Hamiltonian vector field corresponding to a piecewise-linear streamfunction (color gradients).

linearly independent; i.e., vector fields will be generated that are redundant in the sense that they lie in the span of the others. In general, two ways to deal with this situation exist:

The first is to perform distributed orthonormalization. This is the approach taken in e.g. [14], [64], [90], [46], which perform distributed Arnoldi-like iterations to compute Laplacian spectra. The unavoidable disadvantage of approaches of this type is that the computation of inner products inherently requires sums across all of the agents; hence each outer iteration of these algorithms involves an entire consensus problem to compute inner products, and the algorithms are consequently (and necessarily) quite slow.

The second is to accept redundancy. So long as our objective is to generate behaviors satisfying the properties P1, there is no particular reason to believe that a *minimal* set of such behaviors is required. Indeed, it is precisely by relaxing the orthonormality requirement found in spectral algorithms that the algorithms we present obtain their speed.

An example showing the output of the algorithm of Figure 13, compared to that of the previous Figure 8, is given in Figure 14.

3.2.14 Distributed-Infrastructure Routing: Conclusions

We have developed a collection of distributed, consensus-like algorithms by which static infrastructure nodes can synthesize controllers for mobile agents that cause them to circulate throughout an environment without either concentrating their mass in any location, or following paths with contractible loops. This is done first by combining existing algorithms for computing flows and synthesizing controllers that agree with the flows; then by, in a unified fashion, computing controllers and flows together in a symmetric, 2-hop algorithm; and finally by an equivalent 1-hop algorithm that, remarkably, arises from a *directed* Laplacian. The result is a family of linear protocols that converge exponentially to hybrid controllers representing the topology of the environment.

CHAPTER IV

LAGRANGIAN SWARMS

In contrast to the previous chapter, in which we described multiagent systems in terms that rendered agent identities irrelevant, in this chapter we will address swarm representations that explicitly maintain the state of all agents in the system. We will begin by describing multiagent systems as collections of Lagrangian/geodesic systems that interact via holonomic constraints, and then describe approximate formation tracking algorithms that allow a controller, by manipulating just a few "leader" agents, to effect global, approximate formation tracking.

For our purposes, each agent will be a Lagrangian mechanical system. We will initially, following [18], describe these systems in a coordinate-free, differential-geometric framework; however, in later sections, we will, invoking the implicit function theorem, impose local coordinate charts and solve optimal control problems on general nonlinear systems on $\mathbb{R}^n$.

4.1 Constraint-Coupled Mechanical Systems

We consider a collection of mobile agents, comprised of a finite set L of kinematically-controlled *leader* agents that represent the exogeneous input to the system, and a finite set F of *follower* agents that interact with each other and with the leaders via holonomic constraints. Together, these agents are connected in a communication graph G on the vertex set $L \bigcup F$. Each agent $i \in L \bigcup F$ is a geometric mechanical system, with configuration manifold Q_i, and Riemannian metric $\mathbb{G}_i$. Corresponding to each agent $i \in F$, we have a potential field $V_i : Q_i \to \mathbb{R}$, and, for each edge $(i,j) \in E(G)$, holonomic constraints $h_{ij} : Q_i \times Q_j \to \mathbb{R}^{m_{ij}}$ and interagent potentials $V_{ij} : Q_i \times Q_j \to \mathbb{R}$, which give rise to a configuration trajectory $\gamma_i : \mathbb{R}_+ \to Q_i$ for

all agents in $L \bigcup F$, and Lagrange multipliers $\lambda_{ij} : \mathbb{R}_+ \to \mathsf{T}^*(Q_i)$ for all $(i,j) \in F \times (L \bigcup F)$. The control input signals are, for each $i \in L$, a function $u_i : \mathbb{R}_+ \to \mathsf{T}\, Q_i$ that specifies velocities at all times.[1] We will additionally require that the potential fields V, $\{V_{ij}\}_{ij}$ be continuously-differentiable, and that the constraint functions $\{h_{ij}\}_{ij}$ be twice-continuously-differentiable. Then the combined dynamics are represented by the coordinate-free differential-algebraic equation (DAE),

$$\begin{aligned}
\nabla_{\gamma_i'(t)}\gamma_i'(t) &= -\operatorname{grad} V_i(\gamma_i(t)) - \sum_{j \in \mathcal{N}_i} \operatorname*{grad}_1 V_{ij}(\gamma_i(t), \gamma_j(t)) \\
&\qquad + \mathbb{G}_i^\sharp \left[\sum_{j \in \mathcal{N}_i} h_{i,j}^{*,1}(\gamma_i(t), \gamma_j(t)) \lambda_{ij}(t) \right] \quad \forall\ i \in F \\
\gamma_i'(t) &= u_i(t) \qquad \forall i \in L \\
h_{ij}(\gamma_i(t), \gamma_j(t)) &= \mathbf{0} \qquad \forall\ (i,j) \in E(G) \\
\gamma_i'(0) &= \gamma_0
\end{aligned} \tag{70}$$

with initial condition $\gamma_0 \in \mathsf{T}\, Q$ (note that, since tangent vectors implicitly reference their base points, γ_0 contains not just the initial velocity, but also the initial configuration).

Here, we also use the slightly nonstandard notation $f^{*,i}$ to refer to what we will call the *slot pullback* of a function – a partial derivative akin to the *slot derivative*. Precisely, if $f : S_1 \times \cdots \times S_n \to T$, then $f^{*,i}(s_1, \cdots, s_n) = (s \mapsto f(s_1, \cdots, s_{i-1}, s, s_{i+1}, \cdots, s_n))^*(s_i)$. I.e., it is the standard pullback of f, considered as a function only of its i-th argument. Likewise, $\operatorname{grad}_i f(s_1, \cdots, s_i) \triangleq \operatorname{grad}(s \mapsto f(s_1, \cdots, s_{i-1}, s, s_{i+1}, \cdots, s_n))$ denotes the *slot gradient*.

On a subset $U_i \subset Q_i$ of each configuration manifold for $i \in L \bigcup F$, one may define a coordinate chart $\phi_i : U_i \to \mathbb{R}^{n_i}$, and represent the configuration trajectory in coordinates by $q_i \triangleq \phi_i \circ \gamma_i$. In these coordinates, and with some abuse of notation,

[1] Each u_i must also satisfy $\pi_{\mathsf{T}\, Q} \circ \gamma_i \equiv \pi_{\mathsf{T}\, Q} \circ u_i$, where $\pi_{\mathsf{T}\, Q}$ is the bundle projection map. All this means is that the only velocities that can be specified while at a point $\gamma(t)$ are those that actually exist in $T\, \gamma(t)$.

the first equation of the DAE (70) can be written,

$$\begin{aligned}\ddot{q}_i(t) + \sum_{j=1}^{n_i}\sum_{k=1}^{n_i}\Gamma_{i,jk}(q_i(t))\dot{q}_{i,j}(t)\dot{q}_{i,k}(t) = \\ - M_i^{-1}(q_i(t))\frac{dV_i}{dq_i}^T(q_i(t)) \\ - \sum_{j\in\mathcal{N}_i} M_i^{-1}(q_i(t))\frac{\partial V_{ij}}{\partial q_i}^T(q_i(t), q_j(t)) \\ + M_i^{-1}(q_i(t))\left[\sum_{j\in\mathcal{N}_i}\frac{\partial h_{i,j}}{\partial q_i}^T(q_i(t), q_j(t))\lambda_{ij}(t)\right]\end{aligned} \tag{71}$$

for all $i \in F$, where M is the coordinate representation of $\mathbb{G}$ (i.e., it is the mass matrix), and each $\Gamma_{i,jk}$ is a vector of Christoffel symbols of the second kind. The left side of the equation represents coordinate-free acceleration, which includes the effects of Coriolis and centripetal forces; and the right side represents internal, interaction, and constraint forces acting on the system.

4.1.1 Distance Constraints

A common special case of the dynamics (70) arises when agents are particles – i.e., $Q_1 = \cdots = Q_N = \mathbb{R}^n$, for some $n > 0$ (typically 2 or 3), and the constraints $\{h_{ij}\}_{ij}$ that link agents specify interagent distances. In this case, the (shared) configuration manifold is not just a Riemannian manifold but also an inner product space, and the constraints take the form,

$$h_{ij}(\gamma_1, \gamma_2) \triangleq ||\gamma_1 - \gamma_2||^2 - r_{ij}^2 \qquad \forall\ (i,j) \in E(G) \tag{72}$$

with constants $r_{ij} = r_{ji} > 0$ for all i, j. Dropping time arguments, the dynamics (70) are then represented by the DAE, in coordinates,

$$\begin{aligned} m_i\ddot{q}_i &= -\operatorname{grad} V_i(q_i) + \sum_{j\in\mathcal{N}_i}(q_j - q_i)\lambda_{ij} \\ ||q_j - q_i||^2 &= r_{ij}^2 \end{aligned} \tag{73}$$

which must be solved for $\{q_i\}_{i\in L\bigcup F}$ and $\{\lambda_{ij}\}_{(i,j)\in E(G)}$.

4.1.2 From DAEs to ODEs by the Implicit Function Theorem

In subsequent sections, it will be helpful not to work directly with DAEs like (70), which are numerically problematic, but instead with equivalent, lower-dimensional ODE models. The idea is to represent the submanifold $H \triangleq \{(q_1, \cdots, q_N) \in Q_1 \times \cdots \times Q_N \mid h_{ij}(q_i, q_j) = \mathbf{0} \ \ \forall (i,j) \in E(G)\}$ of $Q_1 \times \cdots Q_N$ not by embedding it in $Q_1 \times \cdots Q_N$, but instead by actually constructing the induced affine connection on that submanifold. The result is a coordinate-free ODE of the form

$$\nabla_{\beta'(t)} \beta'(t) = -\operatorname{grad} V(\beta(t)) \tag{74}$$

where $\beta : \mathbb{R}_+ \to H$. The apparent simplicity of (74) is due to the significant complexity now hidden in the affine connection ∇ on H.

Practically, equation (74) is implemented by recourse to a coordinate chart – which, by the Implicit Function Theorem, is guaranteed to exist locally everywhere. This approach comes with the caveat that coordinate charts will have singularities, but in practice we have found that, by a reasonable choice of coordinate chart, one can position singularities away from the regions of the configuration manifold that are of interest.

4.1.3 A Physical Analogue: The Marionette

A physical analogue exists for swarms that behave according to (73), after only mild relaxations, in the form of marionettes like that shown in figure 15. Here, one has a collection of mechanical systems – the various rigid bodies (arms, legs, torso, etc) that comprise the marionette, that are coupled by joints. We will apply the optimal control algorithms presented in the subsequent sections to the example of a marionette.

In our case, the marionette is modeled as a collection of ten point masses $m_1, \cdots, m_{10}$ joined by massless rods and suspended from four massless strings in two dimensions, as illustrated by Figure 17. The free endpoints $p_1, \cdots, p_4$ of the strings can be moved

Figure 15: The marionette

Figure 16: Human subject

kinematically, and the remainder of the model is fully dynamic. The resulting mechanical system has 11 dynamic and 8 kinematic degrees of freedom, for a state space dimension of 30. The strings are in principle represented by inequality constraints on the distance between the points at which their ends are anchored; in our model these constraints are for simplicity relaxed to exponentially-stiffening springs (for a complete treatment of marionette dynamics including inequality constraints, see [52]).

This model is an exact analogue for constraint-coupled multi-agent formations, where agents take the role of point masses, interagent distance constraints take the role of massless rods, and the string model, based on an artificial potential, corresponds exactly to the artificial potentials typically used for connectivity preservation and collision avoidance.

We denote the marionette's configuration in generalized coordinates by $q(t) \in \mathbb{R}^{11}$; specifically, $q(t)$ consists of the position in $\mathbb{R}^2$ of m_1, together with the nine joint angles. Additionally denoting the generalized velocity by $\nu(t) \in \mathbb{R}^{11}$, the dynamics of the state $x(t) = (q, \nu, p)(t)$ of the marionette are summarized,

$$\dot{q} = \nu \tag{75}$$

$$\dot{\nu} = M(q)^{-1}\left(-\sum_{i,j}\Gamma_{ij}(q)\nu_i\nu_j - \nabla U_{\text{grav}}(q) - \nabla U_{\text{spring}}(q,p) - \mu\nu\right) \tag{76}$$

$$\dot{p} = u \tag{77}$$

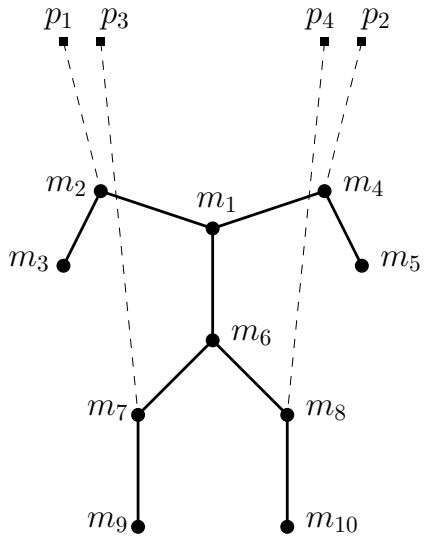

Figure 17: The marionette is modeled as a system of point masses $(m_1, \cdots, m_{10})$ interconnected by massless rods (solid lines) and suspended by strings (dashed lines) from four kinematically-controllable points $(p_1, \cdots, p_4)$.

where $M(q)$ is the mechanical system's mass matrix, each $\Gamma_{ij}(q) \in \mathbb{R}^{11}$ is a vector of Christoffel symbols of the first kind, U_{grav} is the gravitational potential, $\mu \in \mathbb{R}_+$ is a damping coefficient, and $U_{\text{spring}}(q, p)$ is the sum of spring potentials; specifically,

$$U_{\text{spring}}(q, p) = \sum_{(i,j) \in I_{\text{spring}}} k_1 \exp\left[k_2 \left(||\rho_i(q) - p_j||^2 - l_{ij}^2\right)\right] \tag{78}$$

where $I_{\text{spring}} = \{(2,1), (7,3), (4,2), (8,4)\}$ is a set of index pairs summarizing the spring connections, each $\rho_i : \mathbb{R}^{11} \to \mathbb{R}^2$ is the forward kinematic map from the configuration q to the position of the i-th point, l_{ij} is the nominal length of the string connecting m_i and p_j, and $k_1, k_2 \in \mathbb{R}_+$ are chosen constants.

The output map h in this case depends only on the configuration component q of the state, and returns the positions $\rho_1(q), \cdots, \rho_{10}(q)$ of the ten masses $m_1, \cdots, m_{10}$. Our goal is that the output signal y track the corresponding positions $\bar{r} = [\bar{\rho}_1, \cdots, \bar{\rho}_{10}]^T$ of points on a human subject, in an approximate sense that will be introduced in the next sections. More specifically, the reference signal was created by a human dancer who performed the *bhangra* in a motion-capture environment as shown in Figure 16. It consists of the coordinates of the subject's joints as computed by standard

motion-capture software,[2] and projected onto a coronal (or *frontal*) plane.

With the modeling of constraint-coupled, Lagrangian multiagent systems addressed, we now turn our attention to optimal control algorithms for these systems.

4.2 Optimal Control for Approximate Formation Tracking

How should one multiagent formation "mimic," or approximately, subjectively track another? We would like to consider this question in a reasonably general way to begin with that is independent of the particular physical quantities chosen to represent a motion – besides that they can be measured and that they vary in time. Hence, whether they are described by joint angles of an articulated figure, or as the positions in Euclidean space of "marker" points, or by some other quantities, we will view motions simply as signals – albeit signals subject to dynamic constraints.

For instance, consider two motion signals, $y_1 : [0, T] \to \mathbb{R}^m$ and $y_2 : [0, T] \to \mathbb{R}^m$, defined on some time interval $[0, T]$, $T \in \mathbb{R}_+$. Again we ask: How should y_1 and y_2 be compared? An obvious first answer is to use the generalized L^2 metric,

$$d_Q(y_1, y_2) = ||y_1 - y_2||_Q = \sqrt{\int_0^T (y_1(t) - y_2(t))^T Q (y_1(t) - y_2(t)) \, dt} \tag{79}$$

(Q being any symmetric positive definite matrix) and consider motions with smaller distances between them to be "more similar." Alternatively, we might use the angle between the two signals,

$$\angle(y_1, y_2) = \cos^{-1}\left(\frac{\langle y_1, y_2 \rangle_Q}{||y_1||_Q ||y_2||_Q}\right) = \cos^{-1}\left(\frac{\int_0^T y_1^T(t) Q y_2 dt}{\sqrt{\int_0^T y_1^T(t) Q y_1 dt \int_0^T y_2^T(t) Q y_2 dt}}\right) \tag{80}$$

as our measure, and treat motion signals with small angles between them as more similar. However, each of these "dissimilarity measures" leaves much to be desired. For instance, see Figure 18, and suppose that the given signals are joint angle trajectories of a moving arm on a humanoid robot or marionette, and moreover assume

[2] Vicon *VisionIQ* was used.

that the reference signal represents a "wave" motion. Although Signal 1 also represents a "wave" motion, simply dilated and shifted, the qualitatively different signal 2 would be judged more similar using either the L^2 metric (79) or angle (80) as the dissimilarity measure.

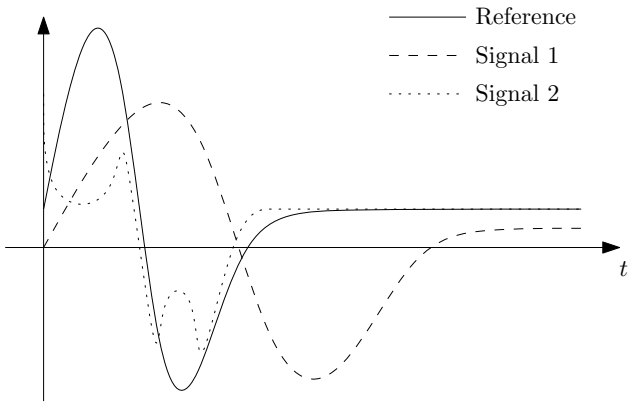

Figure 18: Comparison of three signals: Although subjectively, Signal 1 represents the "same motion" as the reference signal (it is merely shifted, dilated, and scaled) Signal 2 will be judged "more similar" to the reference by either the L^2 metric (79) or angle (80).

A basic problem in both (79) and (80) which is highlighted by Figure 18 is the assumption that the temporal correspondence between different signals is known a priori – that $y_1(t_1)$ should be compared to $y_2(t_1)$ and not $y_2(t_2)$ for all $t_1 \neq t_2$ ($t_1, t_2 \in [0, T]$). Much is assumed about the spatial correspondence as well: (79) presupposes that at any time $t \in [0, T]$ it is preferable that $y_1(t)$ be equal to $y_2(t)$ than to some other value, e.g., a scaled or otherwise transformed value of $y_2(t)$. The angle between signals (80) is somewhat more forgiving – comparing not motions themselves but entire equivalence classes of motions which are the same up to a positive scalar multiple – but nevertheless it cannot tease out any spatial correspondences more complicated than this. Hence, in the sections that follow we will propose a measure of the similarity of motions that makes spatial and temporal correspondence explicit.

4.2.1 Time Warping

A *time warping function* is a function $w : [0, T] \to \mathbb{R}$, $T \in \mathbb{R}_+$ satisfying,

1. w is continuously-differentiable.
2. w is strictly increasing
3. $w(0) = 0$.

Thus w is a bijection, and, moreover, it is a homeomorphism.[3] We will denote the set of all time warping functions (that is, those functions satisfying 1-3 above) by Ω, and the set of derivatives of time warping functions by Ω'; i.e., Ω' is the set of continuous non-negative functions which are zero at at most a finite number of points.

Given a reference signal $r : [0, T] \to \mathbb{R}^p$, $T \in \mathbb{R}_+$, and another signal $y : \mathbb{R} \to \mathbb{R}^p$, the goal of time warping is to find a function w satisfying the above (i.e., $w \in \Omega$) that minimizes the functional $J : \Omega \to \mathbb{R}_+ \cup \{0\}$ defined,

$$J(w) = \int_0^T ||r(\tau) - (y \circ w)(\tau)||_Q^2 d\tau \quad \forall w \in \Omega \ . \tag{81}$$

For instance, consider the following example:

Example 4.2.1. *Suppose* $T = 2\pi$, $r(\tau) = \sin(\frac{\tau^2}{2\pi})$, *and* $y(t) = \sin(t)$ *for all* $\tau \in [0, T]$ *and* $t \in \mathbb{R}$. *Then the optimal warping function is given by,*

$$w^*(\tau) = \frac{\tau^2}{2\pi} \quad \forall \tau \in [0, T]$$

since $(y \circ w)(\tau) = \sin(\frac{t^2}{2\pi}) = r(\tau)$, *so* $J(w^*) = 0 \leq J(w)$ $\forall w \neq w^*$. *Also note that although in this particular example, exact matching is possible (i.e., there exists a* $w \in \Omega$ *such that* $y \circ w = r$*), this is not generally the case.*

[3] An example of a function that satisfies these requirements and is a homeomorphism but not a diffeomorphism would be $x \mapsto x^3$, because its inverse is not differentiable at the origin.

At this point it may be appropriate to point out that another measure of the similarity of signals (usually applied in a stochastic setting to jump processes) which allows for time warping is the *Skorohod metric* [38],

$$d(r,y) = \inf\left\{\epsilon > 0 : \exists\lambda \in \Lambda, \sup_{\tau\in[0,T]} |y(\lambda(\tau)) - r(\tau)| + \sup_{\tau\in[0,T]} |\tau - \lambda(\tau)| \leq \epsilon\right\} \quad (82)$$

where $\Lambda \subset \Omega$ is the set of all strictly increasing functions λ such that $\lambda(0) = 0$ and $\lambda(T) = T$. Some salient differences between (82) and (81) include that we use generalized L^2 norms instead of supremum norms; allow a slightly larger class of time warping functions; and do not directly penalize deviation of the time warping function from the identity function, but rather deviation of their derivatives. Additionally, the computation of (81) involves a single minimization instead of the nested optimization in (82). The strength of the Skorohod metric, which we do not require, is that it results in a complete metric space. An advantage of the cost (81) over (82) for our purposes is that it considers the duration and not just the amplitude of deviations between signals.

We can use the time warping idea to formulate a "time-warped" output-tracking problem as well. We will do this in two ways: First, we will solve the problem when "any" time warping function is allowed; we call this the *nonparametric* time warping problem. We follow this with a look at techniques that fix a particular parametrized form for the time warping function – we address linear functions and, more generally, polynomials – and which optimize over those parameters.

4.2.1.0.3 Tracking with Nonparametric Time Warping

Given a (possibly) nonlinear system of the form,

$$\begin{aligned} \frac{dx_t}{dt}(t) &= f(x_t(t), u_t(t), t) \\ y_t(t) &= h(x_t(t)) \end{aligned} \quad (83)$$

(with $x_t(t) \in \mathbb{R}^n$, $u_t(t) \in \mathbb{R}^m$, $y_t(t) \in \mathbb{R}^p$, and compatible dimensions for the domain and codomain of f and h)[4] we consider, to begin, the optimal control problem,

$$\min_{u \in U_{[0,T]}, w \in \Omega} \left[\int_0^T ||y_t(w(\tau)) - r(\tau)||_Q^2 d\tau + \int_0^{w(T)} ||u_t(t)||_{R_u}^2 dt \right] \tag{84}$$

or equivalently,

$$\min_{u \in U_{[0,T]}, w \in \Omega} \int_0^T \left[||y_t(w(\tau)) - r(\tau)||_Q^2 + w'(\tau)||u_t(w(\tau))||_{R_u}^2 \right] d\tau \tag{85}$$

where $r : [0, T] \to \mathbb{R}^p$ is a reference signal, and $U_{[0,T]}$ is the set of piecewise-continuous functions from $[0, T]$ to $\mathbb{R}^m$.[5] In other words, we would like the output y to "track" the reference signal r as closely as possible. However, note that this differs from the usual tracking problem by the introduction of the time warping function $w : [0, T] \to [0, \infty)$, and also in that we integrate tracking error over *reference time* ("τ time") instead of *system time* ("t time").

Defining $v \triangleq w'$ as the derivative of w, applying the chain rule to (83), and augmenting the state with the time $t = w(\tau)$ we can write the dynamics in reference time as,

$$\frac{d}{d\tau} \begin{bmatrix} x_t \circ w \\ w \end{bmatrix} (\tau) = \begin{bmatrix} v(\tau) f((x_t \circ w)(\tau), (u_t \circ w)(\tau), w(\tau)) \\ v(\tau) \end{bmatrix} \tag{86}$$

or defining $x \triangleq x_t \circ w$, $u \triangleq u_t \circ w$, and $y \triangleq y_t \circ w$, and moreover treating t as a function of time (so in fact $t = w$), we can simplify this notation and combine the above with (85) to state the following standard (Bolza) optimal control problem: Given the system,

$$\frac{d}{d\tau} \begin{bmatrix} x \\ t \end{bmatrix} (\tau) = \begin{bmatrix} v(\tau) f(x(\tau), u(\tau), t(\tau)) \\ v(\tau) \end{bmatrix} \tag{87}$$

$$y(\tau) = h(x(\tau))$$

[4]Note that the subscript t is simply part of the function names x_t, y_t, etc, and is used to distinguish these functions from others to be introduced later.

[5]In (85) and elsewhere we denote $|| \cdot ||_M^2 \triangleq (\cdot)^T M(\cdot)$ and assume $M = M^T \succ 0$, for whichever matrix M is used in the subscript.

with known initial conditions $(x,t)(0) = (x_0, 0)$, minimize the cost functional

$$J_{\text{track}}(u,v) = \int_0^T \left[||y(\tau) - r(\tau)||_Q^2 + v(\tau)||u(\tau)||_{R_u}^2 \right] d\tau \tag{88}$$

over the functions u and v over $U_{[0,T]}$ and Ω', respectively (these functions can be viewed as control inputs to the system).

In fact, however, we will actually use a slightly modified cost that also penalizes large deviations of $w'(\tau)$ from one, both to regularize the problem and to capture the intuition that signals that must be "warped" by a great deal are more dissimilar than those that do not need to be warped as much; this is given below:

$$\begin{aligned} J &: U_{[0,T]} \times \Omega' \to \mathbb{R} \\ J(u,v) &= J_{\text{track}}(u,v) + J_{\text{timewarp}}(v) \\ &= \int_0^T \left[||y(\tau) - r(\tau)||_Q^2 + v(\tau)||u(\tau)||_{R_u}^2 + R_v(v(\tau)-1)^2 \right] d\tau \ . \end{aligned} \tag{89}$$

This, together with (87), is the problem with which we will be interested in this section. Since it is also a Bolza problem, the solution can be found using standard optimal control theory, which we apply in the next section. First, however, we will present a brief discussion of the significance of dynamics to the time warping problem in order to build some intuition:

Example 4.2.2. *Consider the underdamped simple harmonic oscillator with state* $x_t(t) = (x_{t,1}, x_{t,2})(t) \in \mathbb{R}^2$ *and dynamics*

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} x_{t,1} \\ x_{t,2} \end{bmatrix} (t) &= \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{bmatrix} \begin{bmatrix} x_{t,1} \\ x_{t,2} \end{bmatrix} (t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_t(t) \\ y_t(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_t(t) \end{aligned} \tag{90}$$

with $\zeta \in (0,1) \subset \mathbb{R}$, $\omega_0 \in \mathbb{R}_+$, *and suppose that we would like to minimize the infinite-time cost (a modified version of (88)),*

$$\lim_{T\to\infty} \int_0^T \left[\frac{1}{T} ||(y_t \circ w)(\tau) - r(\tau)||_Q^2 + \frac{1}{w(T)} w'(\tau) ||(u_t \circ w)(\tau)||_{R_u}^2 \right] d\tau \tag{91}$$

where the reference signal to be tracked is the sinusoid,

$$r(t) = \cos(\omega_r t) \ \forall t \in [0, \infty) \ . \tag{92}$$

with $\omega_r \in \mathbb{R}_+$. *Moreover for clarity of exposition we will limit our attention to time warping functions of the form*

$$w(\tau) = \xi\tau \quad \forall \tau \in [0, T] \tag{93}$$

for some $\xi \in \mathbb{R}_+$ *(Such* parametric time warping functions *are discussed in more detail in section 4.2.1.1).*

Before proceeding, we point out that in this example the control input that globally minimizes the cost (91) will not be unique. This is, informally, because components of signals that are bounded and occur for finite time "disappear" in the infinite-time average of (91). In fact, if u^* *is a minimizer for (91) (with* ξ *fixed), then so is* $u^* + \tilde{u}$ *for any bounded* $\tilde{u}$ *such that* $\lim_{t\to\infty} \tilde{u}(t) = 0$. *The consequence for this example is that it is only the behavior of the various signals as* $t \to \infty$ *that is of interest.*

We note that the presence of frequencies other than ω_r *in* $u_t \circ w$ *(and hence in* $y_t \circ w$*) increases both terms of (91),*[6] *so* $u_t \circ w$ *and* $y_t \circ w$ *must approach sinusoids with angular frequency* ω_r *as* $t \to \infty$*; without loss of generality (by the previous paragraph) we will assume that they are in fact sinusoids. Also observing that the phase of* $u_t \circ w$ *has no effect on the second term of (91), it must be that* $y \circ w = a\cos(\omega_r t)$ *for some* $a \in \mathbb{R}_+$, *and*

$$u_t(t) = \left(\frac{a}{h(i\omega_r/\xi)} e^{i\frac{\omega_r}{\xi}t} \right) \tag{94}$$

where h *is the transfer function defined by* $h(s) = 1/(s^2 + 2\zeta\omega_0 s + \omega_0^2)$.

[6] *These arguments can be made rigorous using Plancherel's identity for Fourier series and considering a sequence of values for* T *that are multiples of* $\frac{2\pi}{\omega_r}$*; we have omitted this lengthier development for the purposes of our informal discussion.*

It follows that the minimization problem (91) then reduces to,

$$\min_{a,\xi} \lim_{T\to\infty} \left[\frac{Q}{T} \int_0^T (a-1)^2 \cos^2(\omega_r \tau) d\tau + \frac{R_u}{\xi T} \int_0^{\xi T} \left(\frac{a}{h(i\omega_r/\xi)} e^{i\frac{\omega_r}{\xi}t} \right)^2 dt \right]$$
$$= \min_{a,\xi} \left[\frac{Q}{2}(a-1)^2 + \frac{R_u}{2} \left| \frac{a}{h(i\omega_r/\xi)} \right|^2 \right] \ . \tag{95}$$

For any fixed a, this is minimized with respect to ξ when the magnitude of the transfer function $|h(i\omega_r/\xi)|$ is maximized. This occurs when $\xi = \xi^* \triangleq \omega_r/(\omega_0\sqrt{1-\zeta^2})$ – that is, when ξ is chosen so that the resonant frequency of the system with system matrix

$$\xi \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{bmatrix} \tag{96}$$

coincides with the frequency of the reference signal.

The minimizing value of a (which is of less interest) is given by

$$a = \frac{|h(i\omega_r/\xi^*)|}{|h(i\omega_r/\xi^*)| + R_u/Q} \ . \tag{97}$$

The key point demonstrated by this example is that the steady-state effect of time warping is to scale the frequency axis (by the Fourier Dilation Theorem) so that passbands of the system coincide with concentrations of energy in the reference signal. (In fact, this effect is sufficient to overcome certain performance limitations, caused by unstable zero dynamics, that are inherent to the unwarped reference tracking problem; we refer the interested reader to [66] and [4].)

4.2.1.0.4 Optimality Conditions

Theorem 10. *The first order necessary optimality conditions for the minimization of (89) are*

$$\begin{aligned} 2v(\tau)u^T(\tau)R_u + v(\tau)\lambda^T(\tau)\tfrac{\partial f}{\partial u}(x(\tau),u(\tau),t(\tau)) &= \mathbf{0}^T \\ ||u(\tau)||^2_{R_u} + 2R_v v(\tau-1) + \lambda^T(\tau) f(x(\tau),u(\tau),t(\tau)) + \mu(\tau) &= 0 \end{aligned} \qquad \forall \tau \in [0,T] \tag{98}$$

where λ is the solution to the backwards differential equations (101), (102).

Proof : Taking (x,t) to be the state, (u,v) the control input, and (λ,μ) the costate, the Hamiltonian for this problem is,

$$H((x,t),(u,v),(\lambda,\mu),\tau) = L(x,(u,v),\tau) + v\lambda^T f(x,u,t) + \mu v \tag{99}$$

where

$$L(x,(u,v),\tau) = ||h(x) - r(\tau)||_Q^2 + v(\tau)||u||_{R_u}^2 + R_v(v(\tau)-1)^2 \tag{100}$$

and the costate equations are,

$$\begin{aligned}
-\frac{d\lambda}{d\tau}(\tau) &= \frac{\partial H}{\partial x}^T((x,t)(\tau),(u,v)(\tau),(\lambda,\mu)(\tau),\tau) \\
&= \frac{\partial L}{\partial x}^T(x(\tau),(u,v)(\tau),\tau) + v(\tau)\frac{\partial f}{\partial x}^T(x(\tau),u(\tau),t(\tau))\lambda(\tau) \\
&= 2h'(x)^T Q(h(x)-r(\tau)) + v(\tau)\frac{\partial f}{\partial x}^T(x(\tau),u(\tau),t(\tau))\lambda(\tau) \qquad (101) \\
-\frac{d\mu}{d\tau}(\tau) &= \frac{\partial H}{\partial t}((x,t)(\tau),(u,v)(\tau),(\lambda,\mu)(\tau),\tau) \\
&= v(\tau)\frac{\partial f}{\partial t}^T(x(\tau),u(\tau),t(\tau))\lambda(\tau) \qquad (102)
\end{aligned}$$

with $(\lambda,\mu)(T) = 0$. Note that when f is not time-varying, $\frac{\partial f}{\partial t} = 0$, which gives the simplification that $\mu(\tau) = 0\ \forall \tau \in [0,T]$.

In any case, the first order necessary optimality conditions (FONCs) are,

$$\begin{aligned}
&\frac{\partial H}{\partial u}((x,t)(\tau),(u,v)(\tau),(\lambda,\mu)(\tau),\tau) = \\
&\qquad 2v(\tau)u^T(\tau)R_u + v(\tau)\lambda^T(\tau)\tfrac{\partial f}{\partial u}(x(\tau),u(\tau),t(\tau)) = \mathbf{0}^T \\
&\frac{\partial H}{\partial v}((x,t)(\tau),(u,v)(\tau),(\lambda,\mu)(\tau),\tau) = \\
&\qquad ||u(\tau)||_{R_u}^2 + 2R_v v(\tau-1) + \lambda^T(\tau)f(x(\tau),u(\tau),t(\tau)) + \mu(\tau) = 0
\end{aligned} \tag{103}$$

for all $\tau \in [0,T]$. ∎

In fact, these equations (103) can be given the stronger interpretation of stating that the gradient of the functional (89) in the functional space of which (u,v) is an element must be zero; we will take advantage of this interpretation in the later section 4.2.1.2 which describes an algorithm for computing the optimal (u,v).

Example 4.2.3. *The preceding, more general formulation can be used to solve the usual, more specific time-warping problem (81). Given signals y and r to be compared, we define the "signal generator" system,*

$$\dot{x}_t(t) = y'(t) \quad , \quad x_t(0) = y(0) \tag{104}$$

in which case the corresponding augmented system in reference time is

$$\frac{d}{d\tau}\begin{bmatrix} x \\ t \end{bmatrix}(\tau) = \begin{bmatrix} v(\tau)y'(t) \\ v(\tau) \end{bmatrix}. \tag{105}$$

The Hamiltonian is (letting $Q = 1$, $R_v = k \in \mathbb{R}$)

$$H((x,t), v, (\lambda,\mu), \tau) = (x - r(\tau))^2 + k(v-1)^2 + \lambda v y'(t) + \mu v, \tag{106}$$

and we obtain the costate equations

$$-\frac{d\lambda}{d\tau}(\tau) = 2(x - r(\tau)) \tag{107}$$

$$-\frac{d\mu}{d\tau}(\tau) = \lambda(\tau)v(\tau)y''(t(\tau)) \tag{108}$$

with $(\lambda,\mu)(T) = 0$. Finally, the FONC is,

$$\frac{\partial H}{\partial v}((x,t)(\tau), v(\tau), (\lambda,\mu)(\tau), \tau) = \mu(\tau) + \lambda(\tau)y'(t(\tau)) = 0 \tag{109}$$

which converts the problem into a two point boundary value problem.

4.2.1.1 Tracking with Parametric Time Warping

In some situations, we may be interested only in time warping functions with a particular parametric form. One example is linear time-warping functions, which are of special interest since they represent a uniform scaling of the time axis. Another motive for investigating time warping functions with given parametric forms is the discretization of the problem for numerical solution.

To express these ideas, we introduce a parameter vector ξ in some parameter set $\Xi \subset \mathbb{R}^q$, and a *parametrization function* $\phi_v : \Xi \to \Omega'$ which, given a parameter vector,

returns the derivative of a time warping function. Then, we are in fact considering the problem,

$$\min_{\xi,u} J(u, \phi_v(\xi)) \ . \tag{110}$$

In the following subsections, we will first consider parametrization functions that return linear and polynomial time warping functions (whose structure allow them to be treated nicely under the Bolza framework), and then give a more general view of the problem.

4.2.1.1.1 Linear Time Warping

Linear time warping functions are of the form,

$$w(\tau) = \xi\tau \tag{111}$$

or in terms of parametrization functions,

$$\phi_v(\xi)(\tau) = \frac{d}{d\tau}(\xi\tau) = \xi \tag{112}$$

for all $\xi \in \Xi \triangleq \mathbb{R}_+, \tau \in [0, T]$. Letting $v = \phi_v(\xi)$ (so $v(\tau) = \xi \ \forall \tau \in [0, T]$), we can address this problem by augmenting the state with v (rather than treating v as a control input as before) to obtain yet another Bolza problem: Given the system,

$$\frac{d}{d\tau}\begin{bmatrix} x \\ t \\ v \end{bmatrix}(\tau) = \begin{bmatrix} v(\tau)(f(x(\tau), u(\tau), t(\tau)) \\ v(\tau) \\ 0 \end{bmatrix} \tag{113}$$
$$y(\tau) = h(x(\tau))$$

with partially-known initial conditions $(x, t)(0) = (x_0, 0)$, minimize (89) with respect to the initial condition $v(0) = \xi$. Rather than giving a specialized solution to this problem here, we will instead present the more general polynomial time warping problem of which this is a special case. First, however, we would like to point out some features of this minimization problem: Referring to figure 19, we note that this

is a highly nonconvex problem, so without an impractical amount of searching we can at best expect local solutions to the problem. This said, even local solutions offer improvement over the solution to the standard LQ tracking problem, as the time warping still represents additional degrees of freedom that can be exploited to find motions which are "similar" in a sense that is less strict than the usual L^2 norm.

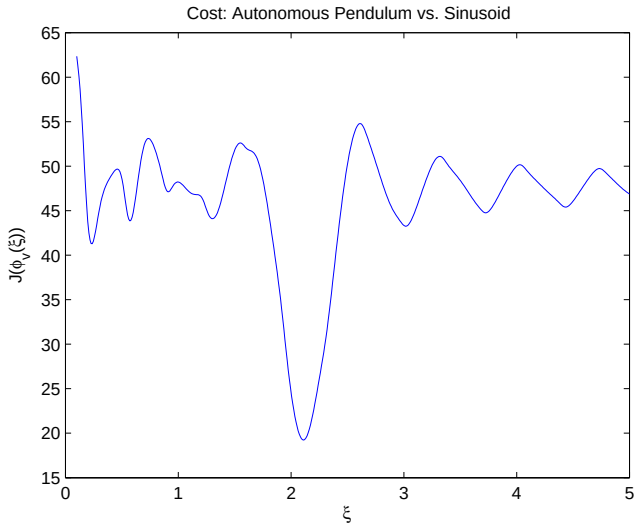

Figure 19: $J(\xi)$ is plotted against the linear time warping parameter $\xi = v(0)$ for the problem in which an autonomous nonlinear pendulum approximately tracks a sinusoid.

4.2.1.1.2 Polynomial Time Warping

Polynomial time warping functions are of the form,

$$w(\tau) = \phi_v(\xi)(\tau) = \sum_{i=1}^{N_v} \xi_i \tau^i \tag{114}$$

for some integer $N \geq 1$, and with discrete parameter vector $\xi = [\xi_1, \ldots, \xi_N] \in \mathbb{R}^{N_v}$ (Note that the requirement that $w(0) = 0$ implies that there is no constant term in the polynomial). As in the linear case in the previous subsection, we will augment the system to obtain a Bolza problem with partially-free initial conditions. The structure of polynomials will give us a particularly convenient way to do this, by representing

polynomials as the output of an autonomous linear system. Differentiating (114), we obtain,

$$w'(\tau) = \sum_{i=1}^{N_v} i\xi_i \tau^{i-1} \ . \tag{115}$$

Functions of the form (115) can be generated by the autonomous system,

$$\frac{d}{d\tau}\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N_v-1} \\ v_N \end{bmatrix} = \begin{bmatrix} v_2 \\ v_3 \\ \vdots \\ v_{N_v} \\ 0 \end{bmatrix} \tag{116}$$

with output $w' = v_1$ and initial conditions,

$$v_i(0) = i!\xi_i \ \ \forall i \in [1, \ldots, N_v] \ . \tag{117}$$

Hence, analogously to (113), we define the augmented system,

$$\frac{d}{d\tau}\begin{bmatrix} x \\ t \\ v_1 \\ \vdots \\ v_{N_v-1} \\ v_N \end{bmatrix}(\tau) = \begin{bmatrix} v_1(\tau)(f(x(\tau), u(\tau), t(\tau)) \\ v_1(\tau) \\ v_2(\tau) \\ \vdots \\ v_N(\tau) \\ 0 \end{bmatrix} \tag{118}$$

$$y(\tau) = h(x(\tau))$$

with partially-known initial conditions $(x, t)(0) = (x_0, 0)$, and seek to minimize (89) with respect to the initial conditions $v_1(0), \ldots, v_N(0)$. In order for t to be a valid time warping function, it is also necessary that this minimization be performed such that the constraint $v_1(\tau) > 0 \ \forall \tau \in [0, T]$ is satisfied.

Theorem 11. *The FONCs for the polynomial time warping problem are given by (122) and*

$$\frac{\partial J}{\partial \xi}^T(\xi) = \text{diag}(1, \tfrac{1}{2}, \tfrac{1}{3!}, ..., \tfrac{1}{N_v!})\nu(0) = 0 \tag{119}$$

where ν is the solution to

$$\begin{aligned} -\frac{d\nu}{d\tau}(\tau) &= \frac{\partial H}{\partial v}^T((x,t,v)(\tau), u(\tau), (\lambda,\mu,\nu)(\tau), \tau) \\ &= \begin{bmatrix} ||u||^2_{R_u} + 2R_v(v_1 - 1) + \lambda^T(\tau)f(x(\tau), u(\tau), t(\tau)) + \mu(\tau) \\ \nu_1(\tau) \\ \cdots \\ \nu_{N_v-1}(\tau) \end{bmatrix} = \mathbf{0} \end{aligned} \tag{120}$$

for all $\tau \in [0,T]$, and with $\nu(T) = \mathbf{0}$.

Proof : For the polynomial time warping problem, the Hamiltonian is (here we identify the costate by (λ,μ,ν), where $\nu(\tau) \in \mathbb{R}^{N_v}\ \forall \tau \in [0,T]$),

$$\begin{aligned} H((x,t,v),u,(\lambda,\mu,\nu),\tau) &= L(x,(u,v_1),\tau) + \lambda^T v_1 f(x,u,t) + \mu v_1 + \nu^T[v_2,\ldots,v_N,0]^T \\ &= ||h(x) - r(\tau)||^2_Q + v_1||u||^2_{R_u} + R_v(v_1-1)^2 + \lambda^T v_1 f(x,u,t) + \mu v_1 + \nu^T[v_2,\ldots,v_N,0]^T \end{aligned}$$

and we obtain the costate equation for ν, (those for λ and μ are the same as given in (101, 102)) given by (120).

This gives us a convenient method for minimizing over $\xi = v(0)$ since,

$$\frac{\partial J}{\partial \xi}^T(\xi) = \text{diag}(1, \tfrac{1}{2}, \tfrac{1}{3!}, ..., \tfrac{1}{N_v!})\frac{\partial J^T}{\partial v(0)}(v(0)) = \text{diag}(1, \tfrac{1}{2}, \tfrac{1}{3!}, ..., \tfrac{1}{N_v!})\nu(0)\ . \tag{121}$$

In other words the gradient of the cost with respect to the polynomial coefficients is simply a scaled version of the initial value of the costate ν.

Additionally, the optimality conditions for the control input are,

$$\begin{aligned} &\frac{\partial H}{\partial u}((x,t,v)(\tau), u(\tau), (\lambda,\mu,\nu)(\tau), \tau) = \\ &\qquad 2v_1(\tau)u^T(\tau)R_u + v_1(\tau)\lambda^T(\tau)\frac{\partial f}{\partial u}(x(\tau), u(\tau), t(\tau)) = \mathbf{0}^T \end{aligned} \tag{122}$$

for all $\tau \in [0, T]$. ∎

Finally, we note that, in principle, since Theorem 11 is a restriction of Theorem 10, it can also be proven as a corollary following the approach of the next section (4.2.1.2), in which case (119) would be seen as the projection of the second equation of (103) onto the subspace of polynomial functions. That said, the introduction of the exosystem (118) is particularly convenient, in that it reduces the problem to one that can be addressed within the standard Bolza framework.

4.2.1.2 The Chain Rule for Parametrization Functions

If the Fréchet derivatives of both the parametrization function ϕ and the cost J [defined in (89)] exist, then we may in fact apply the solution given in section 4.2.1 directly to the discretized problem (110) through a simple application of the chain rule.

Since we will be interested in discretizing not just the time warping function v but also the control input u, at this point we will introduce a second parametrization function $\phi_u : \Sigma \to L^2([0, T], \mathbb{R}^m)$ for some $N_u \in \mathbb{N}$ and parameter space $\Sigma \subset \mathbb{R}^{N_u}$. In other words, given some finite-dimensional $\sigma \in \Sigma$, the function ϕ_u returns a control input function $u \in L^2([0, T], \mathbb{R}^m)$. This yields the fully discretized problem,

$$\min_{\sigma,\xi} J(\phi_u(\sigma), \phi_v(\xi)) \triangleq \min_{\sigma,\xi} J_{\phi_u,\phi_\sigma}(\sigma, \xi) \ . \tag{123}$$

With the interpretation that the partial derivative of the Hamiltonian with respect to the control input (as a function of time) is the gradient of J with respect to the control input (projected onto the dynamical constraint),

$$\nabla_u J(u, v)(\tau) = \frac{\partial H}{\partial u}^T ((x(\tau), t(\tau)), (u(\tau), v(\tau)), (\lambda, \mu)(\tau), \tau)$$
$$\nabla_v J(u, v)(\tau) = \frac{\partial H}{\partial v}^T ((x(\tau), t(\tau)), (u(\tau), v(\tau)), (\lambda, \mu)(\tau), \tau)$$

for all $\tau \in [0,T]$. Then, applying the chain rule to (123), the partial gradients of J_{ϕ_u,ϕ_σ} with respect to σ and ξ are given by the inner products,

$$\nabla_\sigma J_{\phi_u,\phi_v}(\sigma,\phi) = \begin{bmatrix} \langle \nabla_{\sigma_1}\phi_u(\sigma), \nabla_u J(\phi_u(\sigma),\phi_v(\xi))\rangle_{L^2([0,T],\mathbb{R}^m)} \\ \vdots \\ \left\langle \nabla_{\sigma_{N_u}}\phi_u(\sigma), \nabla_u J(\phi_u(\sigma),\phi_v(\xi))\right\rangle_{L^2([0,T],\mathbb{R}^m)} \end{bmatrix} \tag{124}$$

$$\nabla_\xi J_{\phi_u,\phi_v}(\sigma,\phi) = \begin{bmatrix} \langle \nabla_{\xi_1}\phi_v(\xi), \nabla_v J(\phi_u(\sigma),\phi_v(\xi))\rangle_{\Omega'} \\ \vdots \\ \left\langle \nabla_{\xi_{N_v}}\phi_v(\xi), \nabla_v J(\phi_u(\sigma),\phi_v(\xi))\right\rangle_{\Omega'} \end{bmatrix} \tag{125}$$

where the inner products in either Ω' or $L^2([0,T],\mathbb{R}^m)$ are defined,

$$\langle a, b\rangle = \int_0^T a^T(\tau) b(\tau) d\tau \tag{126}$$

for any two functions a, b therein. By solving the state and costate ODEs and evaluating the integrals (126) numerically we thus obtain a principled and general way to discretize the problem.

A technical note: The set Ω' equipped with the inner product (126) is not strictly an inner product space because it contains neither the zero element nor an additive inverse, but $\Omega' \cup (-\Omega') \cup \{\mathbf{0}\}$ is (where here $\mathbf{0}$ denotes the zero function), and it is sufficient for our purposes that Ω' be a subset of an inner product space. Moreover $\Omega' \cup (-\Omega') \cup \{\mathbf{0}\}$ is not a *complete* inner product space (and so not a Hilbert space) since Ω' contains points arbitrarily close to functions that are not strictly increasing. Consequently, safeguards are necessary in optimization algorithms to prevent convergence to functions that are not strictly-increasing (but the term of (89) that penalizes deviation of $w' = v$ from unity tends, as a side benefit, to prevent this).

Example 4.2.4. *Suppose ϕ_u and ϕ_v parametrize control inputs by uniform linear b-splines. In other words, the control inputs u and v are represented by uniformly spaced samples, and linear interpolation is used for the intermediate values. Then,*

$\phi_u(\sigma)$ and $\phi_v(\xi)$ can both be expressed as sums of triangular basis functions, as given below,

$$\phi_u(\sigma)(\tau) = \sum_{i=1}^{K} [\sigma_{1+m(i-1)}, \ldots, \sigma_{mi}]^T \operatorname{tri}\left(\frac{K-1}{T}(\tau - i - 1)\right) \tag{127}$$

$$\phi_v(\xi)(\tau) = \sum_{i=1}^{K} \xi_i \operatorname{tri}\left(\frac{K-1}{T}(\tau - i - 1)\right) \tag{128}$$

where $K \in \mathbb{N}$ is the number of samples used. Then, the functional gradients are given by,

$$(\nabla_{\sigma_i}\phi_u(\sigma,\xi))(\tau) = e_{\operatorname{mod}(i-1,m)+1} \operatorname{tri}\left(\frac{K-1}{T}(\tau - \lceil \tfrac{i}{m} \rceil - 1)\right) \ \forall i \in \{1, \ldots, mK\} \tag{129}$$

$$(\nabla_{\xi_i}\phi_v(\sigma,\xi))(\tau) = \operatorname{tri}\left(\frac{K-1}{T}(\tau - i - 1)\right) \ \forall i \in \{1, \ldots, K\} \tag{130}$$

where e_i denotes the i-th element of the natural basis, and

$$\operatorname{tri}(\tau) = \begin{cases} 1 - |\tau| & \text{if} \quad |\tau| \leq 1 \\ 0 & \text{if} \quad |\tau| > 1 \end{cases}. \tag{131}$$

This gives us the finite-dimensional partial gradient for J_{ϕ_u,ϕ_v} with respect to ξ_i,

$$(\nabla_\xi J_{\phi_u,\phi_v}(\sigma,\xi))_i = \int_0^T \frac{\partial H}{\partial v}((x,t,v)(\tau), u(\tau), (\lambda,\mu,\nu)(\tau), \tau) \operatorname{tri}\left(\frac{K-1}{T}(\tau - i - 1)\right) d\tau \tag{132}$$

for all $i \in \{1, \ldots, K\}$. The expression for $\nabla_\sigma J_{\phi_u,\phi_v}$ is similar.

4.2.2 Output Warping

In the previous sections, we assumed that the spatial correspondence between values of the output signal and those of the reference signal was known, and that only the temporal correspondence needed to be determined. In this section, we will additionally assume that the spatial or output-space correspondence is unknown. This explicitly addresses the fact that it is not just the dynamics of the "mimicking" system that may differ from those of the system that generated the reference motion,

but also spatial constraints and scales – a problem evident even in the prototypical example of a large industrial robot arm asked to imitate a human operator.

To treat the problem of spatial correspondence, we will assume that the reference signal r that we have been considering so far is in fact the composition of two functions: the "actual" reference signal $\bar{r} : [0, T] \to \mathbb{R}^p$, and an "output warping function" $s : \mathbb{R}^p \to \mathbb{R}^p$ of our choosing which transforms values of $\bar{r}$ before they are compared to those of the output signal y. In other words, $r = s \circ \bar{r}$.

More precisely, an *output warping function* $s : \mathbb{R}^p \to \mathbb{R}^p$ is a continuous bijective map with continuous inverse (That is, s is a homeomorphism from $\mathbb{R}^p$ to $\mathbb{R}^p$). We will denote the set of all such functions by $\mathcal{S}$.

We will additionally assume that s has a particular parametric form. This is expressed by saying that s is returned by a parametrization function $\phi_s : \mathcal{C} \to \mathcal{S}$, where $\mathcal{C}$ is a finite-dimensional real vector space of dimension N_s; without loss of generality, we will say $\mathcal{C} = \mathbb{R}^{N_s}$ unless otherwise stated.

With these definitions, we can extend the original cost functional (89) to obtain the new cost functional to be minimized,

$$\begin{aligned}
\bar{J} &: U_{[0,T]} \times \Omega' \times \mathcal{C} \to \mathbb{R} \\
\bar{J}(u, v, c) &= \bar{J}_{\text{track}}(u, v, c) + \bar{J}_{\text{timewarp}}(v) + \bar{J}_{\text{outwarp}}(c) \\
&= \int_0^T \left[||y(\tau) - (\phi_s(c) \circ \bar{r})(\tau)||_Q^2 + v(\tau)||u(\tau)||_{R_u}^2 + R_v(v(\tau) - 1)^2 \right] d\tau \\
&\quad + \bar{J}_{\text{outwarp}}(c)
\end{aligned} \tag{133}$$

where $\bar{J}_{\text{outwarp}}$ is some cost used to penalize "large" output warpings, regularize the problem, and in certain cases enforce constraints; its form will be determined by the choice of ϕ_s and is discussed in more detail later.

Differentiating (133) to find the partial gradient with respect to c, we obtain the

FONC,

$$\nabla_c \bar{J} = -\int_0^T (\nabla_c \phi_s(c))(\bar{r}(\tau))^T Q\left[y(\tau) - \phi_s(c)(\bar{r}(\tau))\right] d\tau + \nabla_c \bar{J}_{\text{outwarp}}(c) = \mathbf{0} \quad (134)$$

which must be satisfied in addition to those given in Section 10. Note that in this equation we assume for notational simplicity that c is a column vector, but we may also use other finite-dimensional real vector spaces, like the matrix-vector pairs of the next section.

4.2.2.0.1 Affine Output Warping

Affine output warping functions are of the form,

$$\phi_s((M, z))(r) = s(r) = Mr + z \quad (135)$$

where $M \in \mathbb{R}^{p \times p}$ is an invertible matrix, and $z \in \mathbb{R}^p$. Hence, for affine output warping functions, $c = (M, z)$, $\mathcal{C} = \mathbb{R}^{p \times p} \times \mathbb{R}^p$, and $N_s = p^2 + p$.

In selecting an appropriate $\bar{J}_{\text{outwarp}}$ for this parametrization, we have two goals:

1. to reward "smaller" transformations – that is, those "close" to the identity transformation in some sense – over "larger" ones.

2. to ensure that s remains a bijection. This means penalizing values of (M, z) for which M is singular or nearly-singular.

The following function achieves these goals:

$$\bar{J}_{\text{outwarp}}((M, z)) = \alpha \frac{[\text{tr}(M^T M)]^p}{\det(M^T M)} + \beta \frac{1}{p} \text{tr}[(M - I)^T (M - I)] + \gamma z^T z \quad (136)$$

where $\alpha, \beta, \gamma \in \mathbb{R}_+ \cup \{0\}$ are scalar coefficients used to weight the relative "importance" of the various parts of the cost.

The first term is inversely proportional to the Gram determinant of the columns of M and so gives a measure of "how singular" M is. The presence of the Frobenius norm $\text{tr}(M^T M)$ in the numerator makes this term somewhat independent of the absolute

scaling of the matrix, since $\frac{\mathrm{tr}((\alpha M)^T(\alpha M))^p}{\det((\alpha M)^T(\alpha M))} = \frac{\mathrm{tr}(M^TM)^p}{\det(M^TM))}$ for all $\alpha \in \mathbb{R}$; that is, it is constant on lines passing through the origin in $\mathbb{R}^{p\times p}$. Furthermore, if $\sigma_1, \ldots, \sigma_p$ are the singular values of M, then,

$$p^p \left(\frac{\sigma_{\min}^2}{\sigma_{\max}^2}\right)^p \leq \frac{[\mathrm{tr}(M^TM)]^p}{\det(M^TM)} = \frac{(\sigma_1^2 + \ldots + \sigma_p^2)^p}{\sigma_1^2 \ldots \sigma_p^2} \leq p^p \left(\frac{\sigma_{\max}^2}{\sigma_{\min}^2}\right)^p \tag{137}$$

so the $p-$th root of this term is always between the (squared) l_2 condition number for M and its reciprocal, up to a constant factor p.

The second term is the squared Frobenius norm of $M - I$, and gives the expected value of $||Mx - x||^2/||x||^2$ if x is a random variable drawn from a uniform distribution on a ball of (any) fixed radius centered at the origin in $\mathbb{R}^p$. Together with the third term, it penalizes "large" transformations – i.e., those that differ substantially from the identity map.

Hence for this problem we have,

$$\begin{aligned}\bar{J}(u, v, (M, z)) = &\int_0^T \left[||y(\tau) - M\bar{r}(\tau) - z||_Q^2 + ||u(\tau)||_{R_u}^2 + R_v(v(\tau) - 1)^2\right] d\tau \\ &+ \bar{J}_{\text{outwarp}}(c)\end{aligned} \tag{138}$$

Combining (138) with (136) and taking partial gradients with respect to M and z we obtain the FONCs expressed by Theorem 12.

Theorem 12. *The additional FONCs for the problem (138) are,*

$$\begin{aligned}\nabla_M \bar{J}(u, v, (M, z)) &= \int_0^T 2Q\left(M\bar{r}(\tau) + z - y(\tau)\right) r^T(\tau) d\tau \\ &\quad + \alpha \frac{2}{\det(M^TM)} \left(-\mathrm{tr}(M^TM)^p M^{-T} + p\,\mathrm{tr}(M^TM)^{p-1} M\right) \\ &\quad + 2\beta \frac{1}{p}\left(M - I\right) \\ &= \mathbf{0}\end{aligned} \tag{139}$$

$$\nabla_z \bar{J}(u, v, (M, z)) = \int_0^T 2Q(M\bar{r}(\tau) + z - y(\tau)) d\tau + 2\gamma z = \mathbf{0} \in \mathbb{R}^p \tag{140}$$

which must be satisfied in addition to (103).

Proof : Please see the appendix.

4.2.2.1 Piecewise Affine Output Warping

A natural extension of affine output warping is piecewise affine output warping, which, with a sufficiently fine subdivision of the region $\mathcal{R} \subset \mathbb{R}^p$ of interest, allows for the representation of arbitrary output warping functions to whatever accuracy is desired. This is a particularly attractive representation for computer implementation.

The essential idea will be that we divide the space $\mathcal{R}$ into some number of p-simplices, and use an affine warping function (as described in the previous section) within each of these, chosen in such a way that the resulting piecewise function is continuous. A new element that this adds to the problem is that, in order to enforce that s remain a bijection, more is required than that each of the individual affine warping functions' "M" matrices be full rank; we must also ensure that the images of their domains remain disjoint.

To begin, let $S = \{S^1, \ldots, S^{|S|}\}$ be a collection of $p-$simplices (which we will refer to as the *input simplices* satisfying,

1. $\bigcup S \supset \mathcal{R}$
2. $\text{int}(S^i) \cap \text{int}(S^j) = \emptyset \quad \forall i \neq j$
3. If y^* is a vertex of S^i and $y^* \in S^j$, then y^* is a vertex of S^j $(\forall y^* \in \mathcal{Y},\ i \neq j)$.
4. For each simplex S^i, there is another simplex S^j sharing p vertices with S^i.

The third condition ensures that there are no "t-junctions" in the mesh and the fourth requires that each simplex share a face with another simplex; these are illustrated in Figure 20. Simplicial tessellations which meet these criteria include the Delaunay tessellation of a collection of randomly-selected points, and the Coxeter-Kuhn-Freudenthal tessellation of a regular grid of p-cubes (for more on simplicial tessellations, see e.g. [21]). We will represent each simplex S^i as the $(p+1)$-tuple

of its vertices in $\mathbb{R}^p$, which we denote $S^i = (S^i_1, \ldots, S^i_{p+1})$ $\forall i \in \{1, \ldots, |S|\}$. Furthermore, let $\text{verts}(S) = \{V^S_1, \ldots, V^S_{|\text{verts}(S)|}\}$ be the unique vertices of S, and $\mathcal{G}^S$ be the graph whose nodes are the elements of $\text{verts}(S)$, and in which an edge exists between V^S_k and V^S_l iff $\exists i \in \{1, \ldots, |S|\}$ s.t. $\{(1-t)V^S_k + tV^S_l : t \in [0,1]\} \subset S^i$. Together with requirement 3 above, this means that the edges of $\mathcal{G}^S$, which we denote "edges($\mathcal{G}^S$)," are in 1-1 correspondence with the unique edges of the simplices in S.

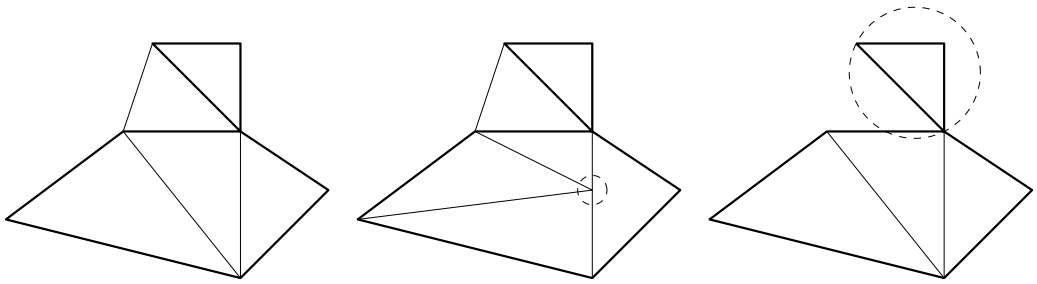

Figure 20: Simplicial tessellations in the plane containing the region outlined in bold satisfying (left) all rules 1-4; (center) all rules except 3 (the offending vertex is circled); and (right) all rules except 4 (the offending simplex is circled).

Next, let $R = \{R^1, \ldots, R^{|S|}\}$ be another collection of $p-$simplices (the *output simplices*) defined by

$$R^i_j = g(S^i_j) \quad \forall (i,j) \in \{1, \ldots, |S|\} \times \{1, \ldots, p+1\} \tag{141}$$

where $g : \mathcal{R} \to \mathbb{R}^p$ is any homeomorphism from $\mathcal{R}$ to some subset of $\mathbb{R}^p$. As for S, we define $\text{verts}(R) = \{V^R_1, \ldots, V^R_{|\text{verts}(R)|}\}$ to be the unique vertices of R; we also define $\mathcal{G}^R$ analogously to $\mathcal{G}^S$. We will consider the elements of $\text{verts}(R)$ to be the parameters defining the output warping function. That is, $c = [(V^R_1)^T, \ldots, (V^R_{|\text{verts}(R)|})^T]^T \in \mathcal{C} = \mathbb{R}^{p|\text{verts}(R)|}$. In other words, holding the input simplices fixed, we optimize over the positions of the vertices of the corresponding output simplices.

Then, we define the output warping function by,

$$s(y) = \Big\{ \; M_i(y - S^i_1) + R^i_1 \quad \text{if} \quad y \in S^i \quad \forall \quad i \in \{1, \ldots, |S|\} \tag{142}$$

where

$$M_i = \begin{bmatrix} R_2^i - R_1^i & R_3^i - R_1^i & \dots & R_{p+1}^i - R_1^i \end{bmatrix} \begin{bmatrix} S_2^i - S_1^i & S_3^i - S_1^i & \dots & S_{p+1}^i - S_1^i \end{bmatrix}^{-1} \tag{143}$$

is a $p \times p$ matrix for each $i \in \{1, \dots, |S|\}$. Equivalently,

$$s(r) = \sum_{i=1}^{p} R_i^{\pi_S(r)} \beta_i(r, S^{\pi_S(r)}) \tag{144}$$

where $\pi_S(r) : \mathcal{R} \to \mathbb{N}$ is the function that, given a point $r \in \mathcal{R}$, returns the index of the simplex in S containing R, and $\beta : \mathcal{R} \times S \to \mathbb{R}^{p+1}$ is the function that, given a point $r \in \mathcal{R}$ and a simplex $S^i \in S$, returns the barycentric coordinates of r in S^i if $r \in S^i$ and $\mathbf{0}$ otherwise. That is, β is defined,

$$\beta(r, S^i) = \begin{cases} \begin{bmatrix} 1 - \mathbf{1}^T M_i (r - S_1^i) \\ M_i (r - S_1^i) \end{bmatrix} & \text{if} \quad r \in S^i \\ \mathbf{0} & \text{if} \quad r \notin S^i \end{cases} . \tag{145}$$

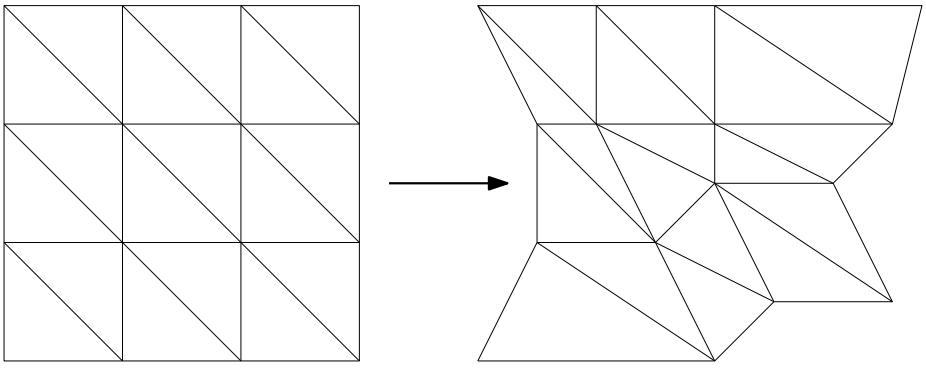

Figure 21: Given the set S of input simplices, the output warping function s is determined by the positions of the vertices of the corresponding output simplices R. This example uses a Coxeter-Kuhn-Freudenthal tessellation of a regular grid of cubes in $\mathbb{R}^2$.

With this representation, a cost which tends to maintain the bijectivity of s is given by,

$$\bar{J}_{\text{outwarp}}(c) = \frac{1}{2} \sum_{(V_i^R, V_j^R) \ \in \ \text{edges}(\mathcal{G}^R)} \left(||V_i^R - V_j^R||_K - ||V_i^S - V_j^S||_K \right)^2 \tag{146}$$

The idea here is that $\mathcal{G}^R$ is a rigid graph, and that by maintaining edge distances we ensure that simplices can neither "collapse" nor "collide." If $\mathcal{G}^R$ is visualized as a network of springs, then (146) gives their overall potential energy.

The partial gradient of $\bar{J}_{\text{outwarp}}$ with respect to each V_i^R is then,

$$\nabla_{V_i^R} \bar{J}_{\text{outwarp}}(c) = \sum_{V_j^R \in \mathcal{N}_{\mathcal{G}^R}(V_i^R)} \frac{||V_i^R - V_j^R||_K - ||V_i^S - V_j^S||_K}{||V_j^R - V_i^R||_K} K(V_j^R - V_i^R) \tag{147}$$

where $\mathcal{N}_{\mathcal{G}^R}(V_i^R)$ is the neighborhood of V_i^R in $\mathcal{G}^R$.

Admittedly, this cost does leave something to be desired, since simplices can collapse with finite energy. Nevertheless, we believe it is useful for its simplicity. One may wish to also apply (136) for each simplex in cases where (146) is not sufficient.

Now, define $c = \begin{bmatrix} (V_1^R)^T & \dots & (|V^R_{\text{verts}(R)|})^T \end{bmatrix}^T$. Letting $i_1(r), \dots i_{p+1}(r)$ be the indexes into verts(S) corresponding to the vertices of the simplex in S containing r, letting $\pi_S(r)$ be the simplex in S containing r, and defining the $p|\operatorname{verts}(S)| \times p$ matrix

$$Z(r) = \begin{bmatrix} I\alpha_1 \\ \vdots \\ I\alpha_{|\operatorname{verts}(S)|} \end{bmatrix} \tag{148}$$

where $\alpha_{i_1(r)} = \beta_1(r, \pi_S(r)), \dots, \alpha_{i_{p+1}(r)} = \beta_{p+1}(r, \pi_S(r))$ and $\alpha_i = 0 \, \forall i \notin \{i_1(r), \dots, i_{p+1}(r)\}$, then the the partial gradient of (133) without the last term $\bar{J}_{\text{outwarp}}$ is given by,

$$\nabla_c(\bar{J} - \bar{J}_{\text{outwarp}})(R) = -2 \int_0^T Z(r(\tau)) Q \left[y(\tau) - (\phi_s(c) \circ r)(\tau) \right] d\tau \ . \tag{149}$$

Hence the partial gradient of $\bar{J}$ with respect to c is simply the sum of (147) and (149).

We apply piecewise affine output warping together with linear time warping in a short example below.

Example 4.2.5. *Suppose we would like the state of an autonomous Van der Pol oscillator to track that of a damped pendulum driven by a fixed-frequency sinusoid, allowing for linear time warping and piecewise affine output warping. That is, the*

system is given by,

$$\frac{d}{dt}\begin{bmatrix} x_{t,1} \\ x_{t,2} \end{bmatrix}(t) = \begin{bmatrix} x_{t,2}(t) \\ \zeta_{vp}(1 - x_{t,1}^2(t)) \end{bmatrix}$$

$$y_t(t) = x_t(t)$$

(with in our case $\zeta_{vp} = 0.9$) and the reference signal r is the solution to,

$$\frac{d}{d\tau}\begin{bmatrix} r_1 \\ r_2 \end{bmatrix}(\tau) = \begin{bmatrix} r_2(\tau) \\ \sin(\tau) - \omega_0^2 \sin(r_1(\tau)) - \zeta_{pend} r_2(\tau) \end{bmatrix}$$

with in our case $\omega_0 = 1$, $\zeta_{pend} = 0.5$, $x(0) = r(0) = [0.1, 0.1]^T$, and $\tau \in [0, T] = [0, 10]$. We will use essentially the same costs introduced earlier, but with some specially-chosen constants; we wish to solve the minimization problem,

$$\min_{\xi,c} \left\{ \begin{array}{l} 0.08 \displaystyle\int_0^T ||y_t(\xi\tau) - (\phi_s(c) \circ \bar{r})(\tau)||^2 d\tau \\ \\ +0.92 \dfrac{1}{2} \displaystyle\sum_{(V_i^R, V_j^R) \in \text{edges}(\mathcal{G}^R)} \gamma_{i,j} \left(||V_i^R - V_j^R|| - ||V_i^S - V_j^S|| \right)^2 \end{array} \right\} \tag{150}$$

where $\gamma_{i,j}$ is the number of simplices containing both V_i^R and V_j^R for all $i, j \in \{1, \ldots, |V^R|\}$, $i \neq j$. Then, performing gradient descent using the gradient given by (147), (149), and (121), we obtain the results shown in Figures 22, 23, and 24.

4.2.3 Example: The Puppet

To demonstrate the application of these ideas, we computed optimal controls for the simplified model of a marionette discussed in Section 4.1.3, which we wished to "mimic" the movements of a human as recorded by a motion capture system.

The optimization takes place with both time- and output- warping. The time warping is nonparametric, and the output warping optimizes over the six scalar parameters defining a two-dimensional affine transformation that is applied to all of

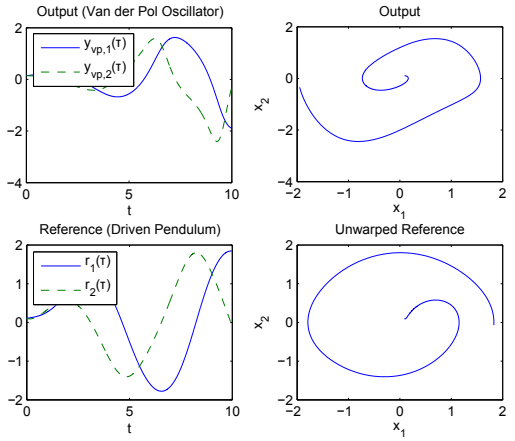

Figure 22: Van der Pol oscillator vs. driven pendulum, before warping. We wish to scale the time axis of the output (top left) and the output space of the reference (bottom right) to align the two signals.

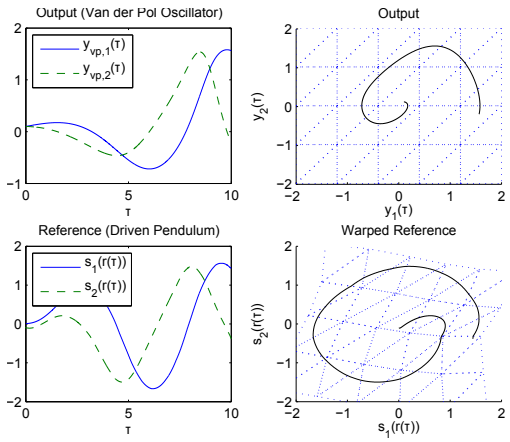

Figure 23: Van der Pol oscillator vs. driven pendulum, after warping. Time warping matches the first part of the Van der Pol oscillator's transient to that of the pendulum (left top, bottom), and output warping rotates and deforms the reference output space to better match the output (right top, bottom).

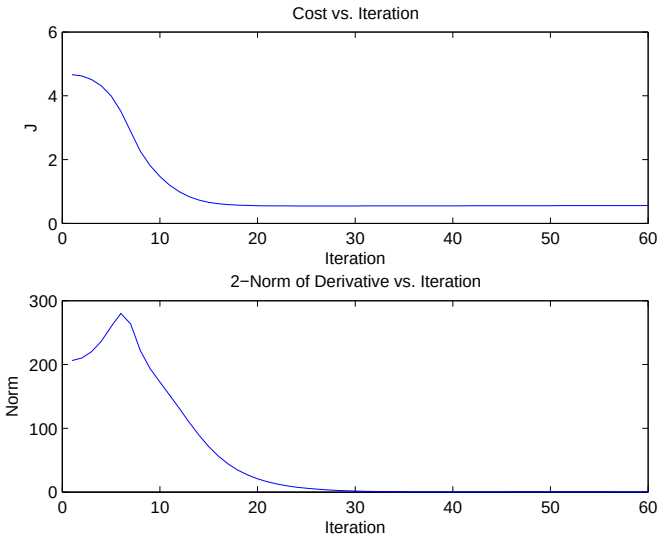

Figure 24: The cost in Example 4.2.5 is reduced (top), eventually reaching a local extremum as evidenced by the reduction of the norm of the derivative to zero (bottom).

the reference points. In other words, for a given warping parameter vector $c =$

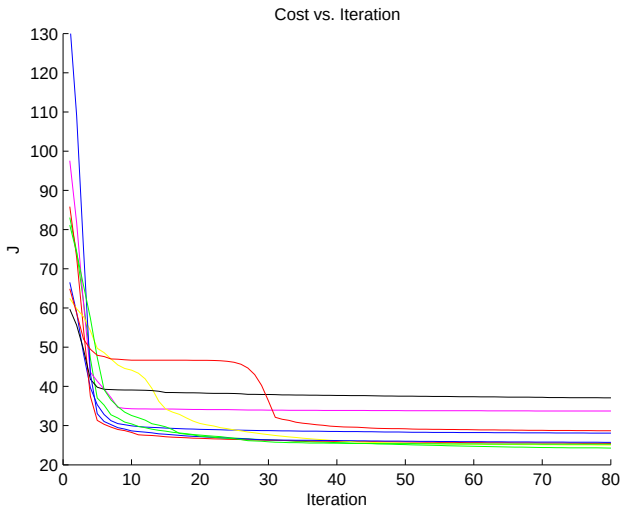

Figure 25: The gradient descent procedure is characterized by a rapid initial descent followed by slower final descent. Each curve corresponds to a different initial guess for the control trajectory; each component of the guess for u takes the form $\alpha\cos(\omega t) + \beta\sin(\omega t)$, with (α, β) chosen uniformly at random from $[-1, 1] \times [-1, 1]$ and ω from $[0, \omega_{\max}]$, where $\omega_{\max}$ is the Nyquist frequency for the sample rate used. The initial guesses for the output warping are realizations of the random variable $I + \frac{1}{5}\epsilon$, where here I denotes the coefficients corresponding to the identity affine transformation and ϵ is drawn uniformly from $[-1, 1]^6$.

$(c_1, \cdots, c_6)$, the corresponding output warping function is given by,

$$s(\bar{r}) = \left(I_{10\times 10} \otimes \begin{bmatrix} c_1 & c_3 \\ c_2 & c_4 \end{bmatrix} \right) \bar{r} + \begin{bmatrix} c_5, c_6, c_5, c_6, \cdots, c_5, c_6 \end{bmatrix}^T \tag{151}$$

where $\otimes$ denotes the Kronecker product.

The results are shown in Figures 25, 26, and 27, where it can be seen that marked improvements in similarity are achieved, both in numerical cost (Figure 25) and in subjective appearance (Figure 26). Both Figure 25 and Figure 27 also illustrate the point that, since the dynamical constraints are nonconvex, the local optimum to which the gradient descent procedure converges depends on the initial condition used to initialize the algorithm. Nevertheless, even for fairly widely-separated initial guesses in the puppet example, similar final costs are obtained.

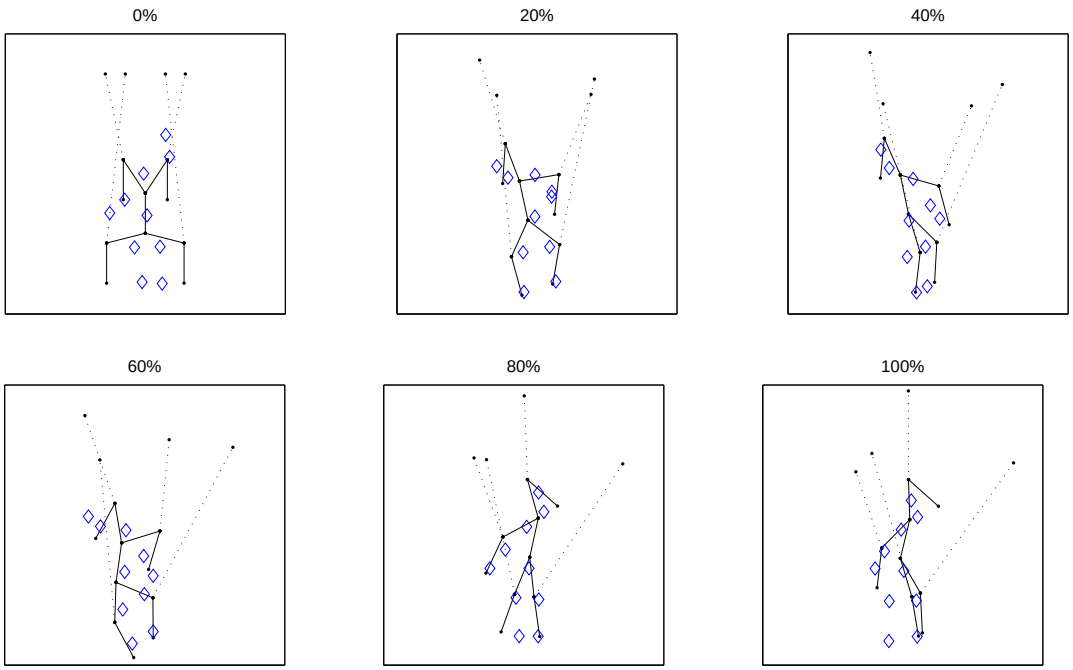

Figure 26: Animation frames showing the lowest-cost imitation of the human subject's bhangra performance by the puppet; percentages are of the total playback time elapsed. The puppet begins hanging in an equilibrium state (at time "0%").

4.2.3.1 Warped Tracking: Contributions

In order to allow one system to "mimic" a reference signal, we have introduced several versions of a modified output tracking problem that also includes time and output warping functions as decision variables. The basic motivation has been that this captures a measure of qualitative similarity which the usual error metrics used in tracking problems (like the generalized L^2 metric) do not.

Time warping in particular takes on a special significance in a controls setting, where, rather than just being a method to compare signals, it actually modifies the dynamics of systems when viewed in "reference time." We have used this fact to reduce the time-warped tracking problem to a standard (Bolza) problem in optimal control, to be solved numerically. The intuition is that we can essentially shift the "passbands" of systems so that they coincide with concentrations of energy in reference signals. Yet although this frequency-domain interpretation is useful to develop

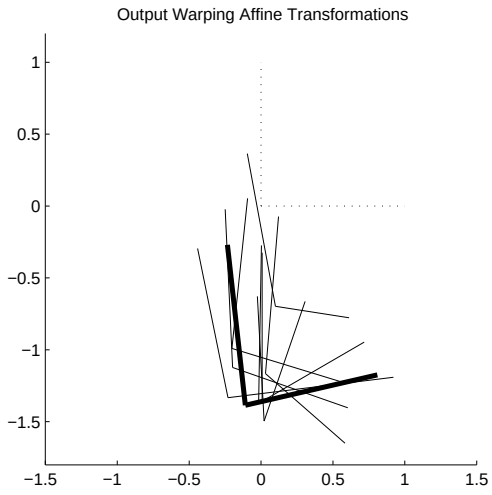

Figure 27: The affine transformations arrived at by the optimization algorithm for different initial guesses (as described in Figure 25) are illustrated as frames in $\mathbb{R}^2$ (solid), along with the identity transformation (dotted). The lowest-cost transformation is drawn in bold.

an intuition for the problem, our development has been entirely time-domain, so that it is applicable to problems with finite time horizons and very general nonlinear systems.

The time-warping problem was studied both when allowing "arbitrary" time warping functions, and when restricting the allowed functions to certain parametric forms. For linear and polynomial time warping functions this allows the problem to be reframed as yet another Bolza problem. For time warping functions with different parametric forms, *parametrization functions* were introduced; this has the advantage of generality but moves slightly outside of the standard Bolza framework.

Likewise, output warping functions having fixed parametric forms were studied; particular attention was paid to affine output warping functions and to a class of piecewise-affine output warping functions defined using a collection of simplices.

In all of the above cases, first order necessary optimality conditions – really gradients – were derived, with the underlying motivation being that this information is necessary for the solution of these optimization problems by first-order numerical

methods - e.g. steepest descent, nonlinear conjugate gradient methods, or quasi-Newton methods. Numerical examples were given for the problems (1) of aligning an autonomous Van der Pol oscillator's output to that of a nonlinear pendulum using linear time warping and piecewise affine output warping, and (2) of controlling a marionette to mimic a human dancer using nonparametric time warping and affine output warping.

The chief limitations of this approach are related to computational tractability, and are common to many problems in numerical optimal control: The gradients, being the solutions to ordinary differential equations, are fairly expensive to compute; and the nonconvexity of the problem means that only local optima are guaranteed. Nevertheless, it is possible to compute local optima which do give results that achieve substantial improvements in qualitative similarity.

CHAPTER V

PREFERENCE LEARNING

5.1 Learning Metric Costs

At the core of preference learning is a collection of empirical, pairwise comparisons. The underlying assumption is that these comparisons reflect an underlying rating function. Hence, given a sequence of pairwise comparisons between points in a Hilbert space, we wish to find (1) a real-valued rating function that is consistent with those preferences, and (2) a global optimum to this function – the best point in the metric space. By solving these two problems we would have recovered what the underlying source for the comparisons is.

Crucially, we would like to be able to determine both the rating function and the point that minimizes it entirely by convex optimization – both so that the resulting problems are computationally efficient, and to ensure that any minima we find are in fact global optima. Although existing approaches like the Support Vector Machine (SVM) methodology of [43] and [29], which we will discuss in Section 5.1.2, do find a rating function as the solution to a convex program, these typically use the so-called *kernel trick*, which introduces nonlinearities that prevent the determination of a minimizer to that function by convex programming. Yet, without the kernel trick, and using the SVM approach, one arrives at linear cost functions that have no unique minima at all. Our contribution, beginning in Section 5.2, is instead a set of convex programs that provide a useful compromise between these extremes, and which only reduce to an SVM classification problem in a particular limiting case. These will allow us to entertain the idea of a unique "best" point in the space, and at the same time determine what it is by convex programming.

5.1.1 Problem Formulation

Given a sequence of pairwise comparisons between points in a Hilbert space, we wish to find a real-valued rating function that is consistent with those preferences. Our motivation is that, given some assumptions about what this function looks like, we would like to find the best possible point.

Formally, let $(X, \langle \cdot, \cdot \rangle)$ be the Hilbert space, and $S \triangleq \{(x_i^1, x_i^2)\}_{i=1}^N \subset X \times X$ the sequence of comparisons; a pair (x_i^1, x_i^2) appears in the sequence S if and only if x_i^1 is preferred to x_i^2. What we seek is a function $f : X \to \mathbb{R}$ such that

$$f(x^1) < f(x^2) \Leftrightarrow (x^1, x^2) \in S \ . \tag{152}$$

That is, we adopt the convention that lower scores are better; hence we will refer to f as a *cost function*.

Moreover, we would like f to minimize some smoothness criterion which we assume will make it generalize well to other points in X besides the ones we have seen in S.

The very general requirement, just introduced, that X be a Hilbert space, will be all that is needed to apply the algorithms that will be developed in the subsequent sections. Examples of Hilbert spaces to which the preference learning of this chapter can be naturally applied include the following:

1. The solution to the time- and output- warped tracking problems of Section 4.2 depends on the choice of symmetric weight matrices Q, R_u, and R_v, output warping parameter vector $\zeta \in \mathcal{C}$,[1] and reference signal $\bar{r}$. Because the vector space of tuples (Q, R_u, R_v, ζ) is real and finite-dimensional, it is a Hilbert space (under any inner product). If the choice of $\bar{r}$ is also restricted to a particular Hilbert space – e.g. L_2 – then, since direct products of Hilbert Spaces[2] are

[1] $\mathcal{C}$ is defined in Section 4.2.2.1. For instance, $\zeta = (\alpha, \beta)$ if (135) is used, or $\zeta = \text{verts}(R)$ if (144) is used.

[2] Here, it is assumed that the inner product for a Hilbert space $A \times B$ is defined, unless otherwise specified, by $\langle (a_1, b_1), (a_2, b_2) \rangle_{A \times B} \triangleq \langle a_1, a_2 \rangle + \langle b_1, b_2 \rangle$.

themselves Hilbert spaces, tuples $(Q, R_u, R_v, \zeta, \bar{r})$ live in a Hilbert space and can be learned under this framework.

2. The space of 1-chains, $C_1(K)$, of Chapter 3, is a finite-dimensional vector space that we have equipped with an inner product; consequently, it too is a Hilbert space, and the input 1-chain $v \in C_1(G)$ can be learned under the preference-learning framework of this chapter.

In the following sections, we investigate a few choices for (1) particular parametric forms for f, and (2) particular smoothness criteria, beginning with the current state of the art. As we finish summarizing the state of the art, our interest in finding the best possible point – a global minimizer of f – will begin to lead us into new territory, starting with Section 5.2.

5.1.2 State of the Art: Support Vector Machines

The Support Vector Machine (SVM) approach to preference learning, which follows the foundational work on SVMs of [15] and [102], was developed in [42] and [43], and has been applied in a number of contexts, including [29]. The following subsections briefly summarize a particular instantiation of this approach; for more detail see [43].

5.1.2.1 Linear cost functions

Initially, we would like to consider linear cost functions. These have the form

$$f(x) = \langle w, x \rangle \tag{153}$$

for some $w \in X$. This will be less of a restriction than it initially appears, because we can achieve nonlinearity by preprocessing, or equivalently via the so-called *kernel trick* (see e.g. [15], [8]).

5.1.2.1.1 Constraints

The first observation, from (152) and linearity of the inner product, is that

$$\left\langle w, x^2 - x^1 \right\rangle > 0 \Leftrightarrow (x^1, x^2) \in S \ . \tag{154}$$

We strengthen the constraints by asserting that, in fact, $\langle w, x^2 - x^1 \rangle > \epsilon$, for some $\epsilon > 0$ and all $(x^1, x^2) \in S$. Since the scale of f is arbitrary (i.e., if f satisfies all pairwise inequality constraints, then so does cf for any $c > 0$), we without loss of generality choose $\epsilon = 1$, and require the stronger constraints,

$$\left\langle w, x^2 - x^1 \right\rangle \geq 1 \quad \forall (x^1, x^2) \in S \ . \tag{155}$$

5.1.2.1.2 Maximizing separation

Subject to the constraints given above, it is traditionally $||w||^2$ that is minimized in the SVM literature. In the case of classification problems, this corresponds to finding the maximum-margin separating hyperplane. In our case, it has another natural justification which we describe in this section.

Consider the quantity

$$\min_{(x^1, x^2) \in S} \left[f(x^2) - f(x^1) \right] \ . \tag{156}$$

If w satisfies the constraints (155), then this number will be positive. In fact, since scale is arbitrary and we are minimizing with respect to $||w||^2$, it will be exactly one (if it were some number greater than one, we could divide f by that number to obtain a new function also satisfying the constraints, corresponding to a w of smaller norm).

Next consider those pairs $(x^1, x^2) \in S$ satisfying $f(x^2) - f(x^1) = 1$. Such points x^1 and x^2 lie on level sets of f which are a distance of $\frac{1}{||w||}$ apart. Therefore, *minimizing* $||w||^2$ *maximizes the distance between the level sets on which lie the least-well-separated points.*

5.1.2.1.3 The Optimization Problem

Combining the minimization problem described in the previous subsection with the constraints (155), we state the following primal optimization problem, which is the

standard *unbiased*[3] SVM classification problem: We wish to find

$$\begin{aligned} &\arg\min_{w} ||w||^2 \\ &\text{s.t. } \langle w, x^2 - x^1 \rangle \geq 1 \quad \forall (x^1, x^2) \in S \end{aligned} . \tag{157}$$

This is a quadratic programming (QP) problem which, in principle, can be solved as-such so long as it is finite-dimensional. The Lagrange dual to this problem – another QP – reveals additional insights however: (1) that the optimal $w \in \text{span}\{x^2 - x^1 : (x^1, x^2) \in S\}$ and so one need only solve for the coefficients of its expansion in terms of this finite basis, and (2) moreover that this expansion is sparse, with only a few terms being nonzero; these elements of $\{x^2 - x^1 \mid (x^1, x^2) \in S\}$ which have nonzero coefficients are called the *support vectors*. For more see [89]. For our purposes it is enough to note (1) that because $w \in \text{span}\{x^2 - x^1 \mid (x^1, x^2) \in S\}$, (157) can be found as the solution to a finite-dimensional problem whose dimensionality is the number of comparisons, regardless of the dimensionality of X (which can even be infinite-dimensional), and (2) that the optimization problem depends only on the inner products between the various elements of $\{x^2 - x^1 \mid (x^1, x^2) \in S\}$. This finite-dimensional dual problem can be solved by general-purpose QP codes, or by special-purpose SVM solvers (which are generally for the more common *biased* SVM classification problem) by augmenting the problem with additional instances in a symmetric way which ensures that the solution hyperplane nevertheless passes through the origin; for more on this latter approach see e.g. [43], [29].

5.1.2.2 Kernelization

The basic idea behind the so-called *kernel trick* (see e.g. [15], [8]) is that, rather than finding a linear classifier for instances directly, one can first map the points to a higher-dimensional inner product space F (called the *feature space*) using a nonlinear

[3]Here, *unbiased* means that the separating hyperplane must pass through the origin. This is as opposed to the *biased* case, where not just linear but also affine varieties are allowed.

embedding $\Phi : X \to F$. Moreover since only inner products between elements of this space need be computed, the embedding need not be explicitly constructed; rather, the function $\kappa : X \times X \to \mathbb{R}$ defined by,

$$\kappa(x^1, x^2) \triangleq \left\langle \Phi(x^1), \Phi(x^2) \right\rangle \tag{158}$$

is all that is actually needed. *Mercer's theorem* establishes conditions under which a Φ and F exist such that a given function from $X \times X$ to $\mathbb{R}$ can be written in the form (158). Specifically, for any arbitrary finite list $x_1, \cdots, x_M \in X$ of instances, the kernel matrix $[\kappa(x_i, x_j)]_{i,j} \in \mathbb{R}^{M \times M}$ must be symmetric positive definite and so a Gramian.

Early work in preference learning (e.g. [42]) simply applied the kernel trick to the classification problem described in section 5.1.2.1. However, as noted by [42], since this implicitly passes *differences* between points through Φ and not the points themselves, this does not preserve the structure of an assumed underlying cost function; i.e.,

$$\left\langle w, \Phi(x^2 - x^1) \right\rangle > 0 \not\Longleftrightarrow \left\langle w, \Phi(x^2) - \Phi(x^1) \right\rangle > 0 \tag{159}$$

and so it is incorrect using this approach to assume that there exists a cost function of the form

$$f(x) = \langle w, \Phi(x) \rangle \ . \tag{160}$$

The work [43] resolves this to recover a cost function which does have the form (160) by classifying not differences between points (elements of X) but rather pairs thereof (elements of $X \times X$), while simultaneously insisting on a particular form for the kernel; this is also the approach used in [29]. In [43] it is shown that a cost function of the form (160) is obtained by using a kernel function $K : (X \times X) \times (X \times X) \to \mathbb{R}$ defined by,

$$K(x^1, x^2; x^3, x^4) \triangleq \kappa(x^1, x^3) - \kappa(x^1, x^4) - \kappa(x^2, x^3) + \kappa(x^2, x^4) \tag{161}$$

in the resulting classification problem, where κ is the kernel function corresponding to the embedding Φ in (160). The kernel K is referred to in [29] as the *Herbrich's Kernel* attached to κ.

This results in the cost function,

$$f(x) = \sum_{i=1}^{N} \alpha_i \left(\kappa(x_i^2, x) - \kappa(x_i^1, x) \right) \tag{162}$$

where the coefficients $\{\alpha_i\}_{i=1}^N \subset \mathbb{R}$ are the solution to the dual problem to (157).

5.1.2.2.1 Limitations

In the remainder of this chapter, motivated by the desire to compute globally-optimal alternatives by convex optimization, we will investigate *metric cost models*, and find that these cannot be addressed efficiently within the SVM framework. This will motivate the construction of Chebyshev estimators. These also employ convex optimization and the idea of maximizing robustness, but only reduce to SVMS in particular special cases. (A more detailed discussion of the incompatibilities that exist between metric cost models and standard Support vector Machine approaches has been deferred to the Appendix.)

5.2 Metric Preference Learning: A Convex Formulation

In this section, we begin to introduce our contributions to preference learning. These include novel, convex problem formulations; an asymptotic "observer" for human preferences that converges under a persistent-excitation assumption; and algorithms that simplify preference data based on purely graph-theoretic considerations. We begin by considering these graph-based simplifications, before introducing metric cost models themselves and their accompanying optimization problems.

5.3 The Preference Graph

The *preference graph* $\mathcal{G} = (V, S)$ corresponding to the comparison sequence S is the directed graph whose vertex set $V = \{x_1^1, x_1^2, \cdots, x_N^1, x_N^2\} \subset X$ is the collection of all

unique points that have been compared, and whose edge set is S. We will index the vertices as $V = \{x_1, \cdots, x_M\}$, where $M \leq 2N$ is the cardinality of V.

If (152) is to hold with strict inequality, then we note immediately that the graph $\mathcal{G}$ must be *acyclic*, and thus represent a *partial order*. When nonstrict inequalities are allowed, however, then we may permit cycles, and moreover $\mathcal{G}$ can be replaced by a smaller, equivalent acyclic graph. This has the practical significance of allowing redundant constraints to be eliminated on purely graph-theoretic grounds, thereby speeding up later optimization steps. This is constructed, following [6], in the following way:

A *cell* is defined to be an equivalence class of vertices; two vertices $v_1, v_2 \in V$ belong to the same cell (denoted $v_1 \sim v_2$) if and only if there exist directed paths in $\mathcal{G}$ from v_1 to v_2 *and* from v_2 to v_1. The quotient graph $\mathcal{G}/\sim$ is the directed acyclic graph whose vertices are these equivalence classes, and in which the directed edge (C_1, C_2) exists between two cells C_1 and C_2 whenever there exist vertices $v_1 \in C_1$ and $v_2 \in C_2$ such that there is a directed path in $\mathcal{G}$ from v_1 to v_2.

Since any two vertices in the same cell must by (152) have the same cost, one may optimize using only the constraints represented by the edges of this quotient graph, and discard the rest. Hence without loss of generality we will assume that $\mathcal{G}$ is acyclic; when it is not it should be understood that we will actually work with $\mathcal{G}/\sim$.

Additional constraints can be eliminated via the *transitive reduction*. Formally, using Aho's definition [6], $\mathcal{G}^t$ is the transitive reduction of a graph $\mathcal{G}$ if,

1. there is a directed path from vertex u to vertex v in $\mathcal{G}^t$ if and only if there is a directed path from u to v in $\mathcal{G}$, and

2. there is no graph with fewer arcs than $\mathcal{G}^t$ satisfying condition 1.

In the case of a directed acyclic graph, the reduction $\mathcal{G}^t$ (which is unique) is a subgraph of $\mathcal{G}$. It was shown in [6] that computation of the transitive reduction is of the

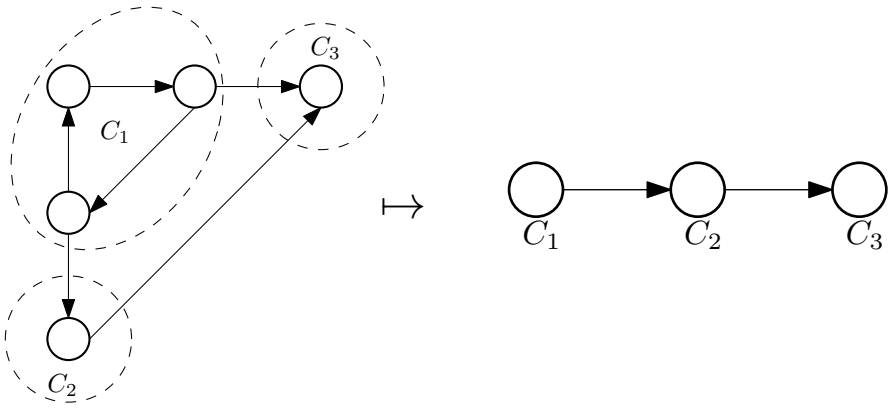

Figure 28: The original preference graph $\mathcal{G}$ (left), and the corresponding transitively-reduced quotient graph, $(\mathcal{G}/\sim)^t$ (right).

same complexity as transitive closure, and hence matrix multiplication; thus, the transitive reduction can be found in $O(n^{\log_2 7})$ steps using Strassen's algorithm [91], or, in principle, $O(n^{2.376})$ steps using the Coopersmith-Winograd algorithm [25]. (See, e.g., [35], [77]). Moreover, if $\mathcal{G}$ contains cycles, then the algorithm given in [6] can compute $(\mathcal{G}/\sim)^t$ with the same complexity.

In short, by working with the transitive reduction of the quotient graph, we are able to eliminate redundant constraints on purely graph-theoretic grounds, before even knowing the form of the cost function f. The reduction is illustrated by Figure 28.

5.4 Metric Costs

Colloquially, when comparing various alternatives, we often speak of options as being "closer to what we would like," or of being "far from perfect." Motivated by this everyday use of geometric language, in [58] we considered *metric costs*, which have the form,

$$f(x) = ||x - \bar{x}||^2 \ . \tag{163}$$

In short, it is assumed that there exists some single best point $\bar{x}$ in X, and one alternative is preferred over another if and only if it is closer to that point.

What does an individual response (x^1, x^2) tell us about the location of $\bar{x}$? Simply, the following are equivalent:

1. $(x_i^1, x_i^2) \in S$
2. $f(x^1) \leq f(x^2)$
3. $\langle x_i^2 - x_i^1, \bar{x} \rangle - \frac{1}{2} \langle x_i^2 - x_i^1, x_i^2 + x_i^1 \rangle < 0.$

In words, each comparison constrains $\bar{x}$ to lie within a particular halfspace of X. Defining,

$$d_i \triangleq x_i^2 - x_i^1 \tag{164}$$

$$\mu_i \triangleq \frac{1}{2}\left(x_i^1 + x_i^2\right) \tag{165}$$

$$b_i \triangleq \langle d_i, \mu_i \rangle \ , \tag{166}$$

the totality of what we know, then, about where $\bar{x}$ might lie is summarized by the inclusion over all the comparison halfspaces,

$$\bar{x} \in P \triangleq \bigcap_{i=1}^{N} \{x \mid \langle d_i, x \rangle - b_i < 0\} \ . \tag{167}$$

The set P, if it is bounded, is a polytope in X. In [58], we stated this system of inequalities and gave an asymptotic observer that $\bar{x}$ under certain assumptions. Here, we ask another question: Out of all the points in this polytope, which is "best?"

When P is bounded, we propose to select $\bar{x}$ as the *incenter* or *Chebyshev center* of the polytope,

$$\bar{x} = \arg\min_x \max_i \frac{1}{||d_i||} \left(\langle d_i, x \rangle - b_i\right) \tag{168}$$

which is the point that is maximally far away from the closest constraint plane, as illustrated by Figure 29. In other words, when P is nonempty, $\bar{x}$ is the point that can be perturbed as much as possible without contradicting any of the preferences expressed in S; and when P is empty, it is the "compromise" point whose worst constraint violation is minimal.

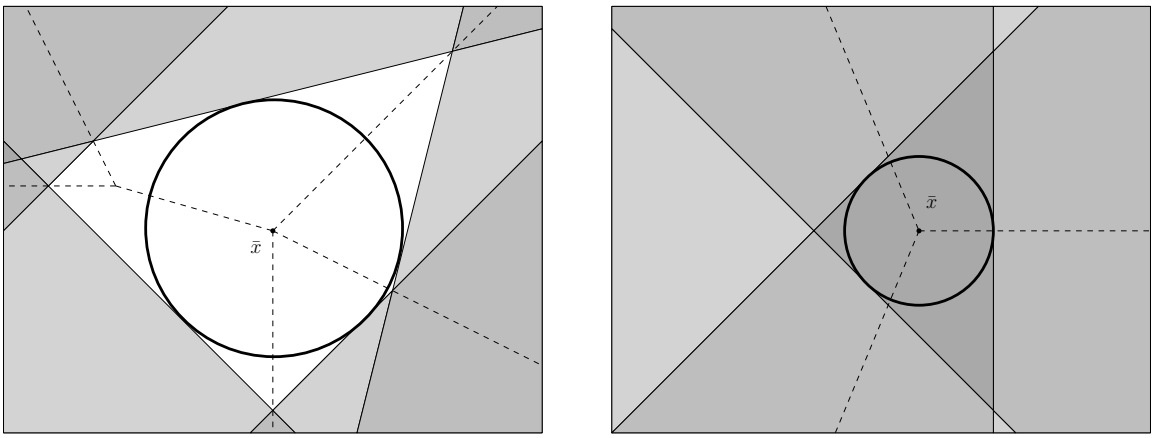

Figure 29: Two examples for $X = \mathbb{R}^2$. Shades of gray indicate the number of violated constraints (points in darker regions violate more constraints), and discontinuities in the derivative of the piecewise-linear function $x \mapsto \max_i \frac{1}{||d_i||} (\langle d_i, x \rangle - b_i)$ are indicated by dashed lines. In the first example (top), $P \neq \emptyset$ (white region), and $\bar{x}$ is its incenter, the point maximally far away from the closest of the constraint surfaces (thin, solid lines) - i.e., it is the center of the largest inscribed sphere (thick, solid curve). In the second example (bottom), $P = \emptyset$, and the resulting optimum, $\bar{x}$, is the point whose worst constraint violation is minimal.

Note that with the definition (168), if the constraints are feasible (i.e., if $P \neq \emptyset$), then $\bar{x} \in P$. This can be viewed as minimizing the ∞-norm of the vector of constraints. Additionally, $\bar{x} \in \text{aff}\,\{x_1^1, x_1^2, \cdots, x_N^1, x_N^2\}$ and hence we need only solve for the coefficients of an expansion in terms of this basis (see Theorem 13). Furthermore, this minimization problem has a sensible solution even when P is empty; it is the point whose worst constraint violation is as small as possible.

This can be rewritten in epigraph form as the constrained minimization problem,

$$(\bar{z}, \bar{x}) = \underset{(z,x)}{\arg\min}\, z \tag{169}$$

$$\text{s.t.} \quad ||d_i|| z \geq \langle d_i, x \rangle - b_i$$

which is always feasible (but possibly unbounded), and satisfies $\bar{z} > 0 \iff P = \emptyset$.

Theorem 13. *If (168) has a global minimizer, then it has a global minimizer in the affine subspace,* aff $\{x_1^1, x_1^2, \cdots, x_N^1, x_N^2\}$.

Proof : Let x be a global minimum to (168), and $\bar{x}$ be the projection of x onto the

affine subspace, aff $\{x_1^1, x_1^2, \cdots, x_N^1, x_N^2\}$; i.e., $\bar{x} = x + \delta$ with $\delta \perp \text{span}\{d_1, \cdots, d_N\}$. Then for all $i \in \{1, \cdots, N\}$, since $\langle d_i, \delta \rangle = 0$ and by linearity of the inner product, $\frac{1}{||d_i||} \langle d_i, \bar{x} \rangle - b_i = \frac{1}{||d_i||} \langle d_i, x \rangle - b_i$, and hence the value of the objective function in (168) is the same at either x or $\bar{x}$. ∎

5.4.1 Direct Solution

The minimization problem (169) can be rewritten, when X is finite-dimensional, in matrix form as,

$$\begin{aligned} &\arg\min_{(z,x)} \begin{bmatrix} 1 & \mathbf{0} \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} \\ &\text{s.t.} \begin{bmatrix} -||d_1|| & d_1^T \\ \vdots & \vdots \\ -||d_N|| & d_N^T \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} \leq \begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix} . \end{aligned} \tag{170}$$

This is a linear program (LP), which can be solved directly by general-purpose LP codes provided $\dim(X)$ is not too large.

5.4.2 Instance Vector Expansion

Since $\bar{x} \in \text{aff}\{x_1^1, x_1^2, \cdots, x_N^1, x_N^2\}$, the optimization problem (168) can be solved as a finite-dimensional problem even when X is not finite-dimensional, by expanding $\bar{x}$ in terms of a finite-dimensional basis, as described by the following theorem:

Theorem 14. *The point*

$$\bar{x} = \sum_{k=1}^{N} \bar{c}_k d_k + x^* \tag{171}$$

solves the optimization problem (168), where

$$x^* = \arg\min_x \left\{ ||x||^2 \mid x \in \text{aff}\left\{x_1^1, x_1^2, \cdots, x_N^1, x_N^2\right\}\right\} \ , \tag{172}$$

and $\bar{c}$ is found by solving

$$(\bar{z}, \bar{c}) = \arg\min_{(z,c)} z$$
$$\text{s.t.} \quad G^{dd}c - Dz \leq \beta \ , \tag{173}$$

with $D = (||d_1||, \cdots, ||d_N||)$, $\beta \in \mathbb{R}^N$ defined by

$$\beta_i \triangleq \langle d_i, \mu_i \rangle \tag{174}$$

and $G^{dd} \in \mathbb{R}^{N \times N}$ being the Gramian,

$$G^{dd} \triangleq \begin{bmatrix} \langle d_1, d_1 \rangle & \cdots & \langle d_1, d_N \rangle \\ \vdots & \ddots & \vdots \\ \langle d_N, d_1 \rangle & \cdots & \langle d_N, d_N \rangle \end{bmatrix} \ . \tag{175}$$

Proof : Defining x^* by (172), one can write any x in the affine span of the data in the form (171). Substituting the expansion (171) into (169) and noting that by Hilbert's Projection Theorem $x^* \perp d_i$ for all $i \in \{1, \cdots, N\}$, one obtains (173). ∎

Remark 1. *We also note at this point that (171) can be written,*

$$x = \sum_{k=1}^{M} \left(\text{indeg}_c(x_k) - \text{outdeg}_c(x_k)\right) x_k + x^* \tag{176}$$
$$\triangleq \sum_{k=1}^{M} \xi_k x_k + x^* \tag{177}$$

by treating c as a vector of edge weights to the preference graph, and denoting the weighted in- and out-degrees of a given node x_k by $\text{indeg}_c(x_k)$ and $\text{outdeg}_c(x_k)$ respectively. Precisely,

$$\text{indeg}_c(x_k) \triangleq \sum_{i | x_i^2 = x_k} c_i \tag{178}$$
$$\text{outdeg}_c(x_k) \triangleq \sum_{i | x_i^1 = x_k} c_i \ . \tag{179}$$

Remark 2. *Moreover, β can be written,*

$$\beta_i = e_i^T G^{\mu d} e_i \ , \tag{180}$$

where $G^{\mu d} \in \mathbb{R}^{N \times N}$ is the cross-Gramian

$$G^{\mu d} \triangleq \begin{bmatrix} \langle d_1, \mu_1 \rangle & \cdots & \langle d_1, \mu_N \rangle \\ \vdots & \ddots & \vdots \\ \langle d_N, \mu_1 \rangle & \cdots & \langle d_N, \mu_N \rangle \end{bmatrix} \tag{181}$$

and e_i denotes the i-th column of the identity matrix.

Remark 3. *Note that this problem depends only on inner products of the various d_i and u_i vectors, and hence the problem can be solved even when X is infinite-dimensional. Precisely, $\frac{N(N+1)}{2} + N^2 \sim O(n^2)$ inner products must be computed to build the matrices G^{dd} and $G^{\mu d}$, where N is the number of comparisons. Alternatively, the relevant matrices can also be produced directly from inner products of elements of S, as*

$$G^{dd} = K^{22} - K^{21} - K^{12} + K^{11} \tag{182}$$

$$G^{\mu d} = \frac{1}{2}(K^{22} + K^{21} - K^{12} - K^{11}) \tag{183}$$

where each matrix $K^{lm} \in \mathbb{R}^{N \times N}$ is defined by

$$K_{ij}^{lm} = \left\langle x_i^l, x_j^m \right\rangle \tag{184}$$

and can be built by indexing into the single Gramian (or kernel) matrix $K \in \mathbb{R}^{M \times M}$ defined,

$$K_{ij} = \langle x_i, x_j \rangle \ . \tag{185}$$

Moreover, $D = (\sqrt{G_{11}^{dd}}, \sqrt{G_{22}^{dd}}, \sqrt{G_{33}^{dd}}, \cdots, \sqrt{G_{NN}^{dd}})$.

Finally, $\bar{x}$ can be reconstructed using (171) and

$$x^* = \sum_{i=1}^{M} \alpha_i x_i \tag{186}$$

$$\alpha = \frac{1}{\mathbf{1}^T K^\dagger \mathbf{1}} K^\dagger \mathbf{1} \tag{187}$$

where $K^\dagger$ denotes the Moore-Penrose pseudoinverse of K, and $\mathbf{1} = (1, 1, \cdots, 1) \in \mathbb{R}^M$.

In particular, the costs of the presented instances can be reconstructed as,

$$f(x_k) = (e_k - \xi - \alpha)^T K (e_k - \xi - \alpha) \tag{188}$$

where ξ is related to c by (176), (178), and (179).

5.4.2.1 Unbounded Case: The minimax-rate problem

When P is nonempty but unbounded, we ask a slightly different question: What is the "point at infinity," or *direction*, that is best? More precisely, what we seek in this case is a unit vector

$$\bar{v} = \underset{v \in X | ||v|| = 1}{\arg\min} \lim_{t \to \infty} \frac{1}{t} \left[\max_i \frac{1}{||d_i||} \left(\langle d_i, tv \rangle - b_i \right) \right] \tag{189}$$

$$= \underset{v \in X | ||v|| = 1}{\arg\min} \max_i \frac{1}{||d_i||} \langle d_i, v \rangle \tag{190}$$

or equivalently,

$$(\bar{p}, \bar{v}) = \underset{v \in X, p \in \mathbb{R}}{\arg\min} p \tag{191}$$

$$\text{s.t.} \quad \begin{cases} ||d_i|| p & \geq & \langle d_i, v \rangle & \forall i \in \{1, \cdots, N\} \\ ||v||^2 & \leq & 1 \end{cases}.$$

As before, an instance vector expansion is possible:

Theorem 15. *Letting $v = \sum_{k=1}^N c_k d_k$, the optimization problem (191) is equivalent to*

$$(\bar{p}, \bar{c}) = \underset{(p,c)}{\arg\min} p \tag{192}$$

$$\text{s.t.} \quad \begin{cases} G^{dd} c - Dp & \leq & 0 \\ c^T G^{dd} c & \leq & 1 \end{cases}$$

with the matrices G^{dd} and D as defined in the previous subsection.

Proof : The proof takes the form of theorem 14's. ∎

The problem (192) is a finite-dimensional second-order cone program (SOCP), which can be solved efficiently.

The cost function for the unbounded case arises from a similar limit process to (190), as

$$f(x) = \lim_{t\to\infty} \left(\frac{1}{t} ||x - vt||^2 - t \right) \tag{193}$$

$$= \lim_{t\to\infty} \left[\frac{1}{t} \left(||x||^2 - 2\langle x, vt \rangle + ||vt||^2 \right) - t \right] \tag{194}$$

$$= -2\langle x, v \rangle \tag{195}$$

which can be evaluated at the instances as,

$$f(x_k) = -2e_k^T K\xi \ . \tag{196}$$

5.4.3 QP Form and Relation to SVMs

When $\operatorname{int} P$ is nonempty - as is almost always true in the unbounded case - the minimization problem (190) can be rewritten as an equivalent quadratic program (QP), which will make the relationship to the usual SVM approach very clear. In fact, (190) is equivalent to a particular SVM classification problem (which differs from but is related to that studied in e.g. [29] and [43]).

Defining,

$$w = \frac{1}{p} v \tag{197}$$

and restricting our attention to negative values for p (since when $\operatorname{int} P$ is nonempty, $p^* < 0$), we note that

$$\arg\min p = \operatorname{argmax} p^2 = \arg\min \frac{1}{p^2} = \arg\min ||w||^2 \ . \tag{198}$$

Additionally, the constraints in (191) can be replaced by,

$$\left\langle \frac{d_i}{||d_i||}, w \right\rangle \geq 1 \tag{199}$$

which results in the standard unbiased SVM problem,

$$\begin{aligned} \bar{w} &= \arg\min_{w} ||w||^2 \\ \text{s.t.} \quad &\left\langle \frac{d_i}{||d_i||}, w \right\rangle \geq 1 \quad \forall i \in \{1, \cdots, N\} \ . \end{aligned} \tag{200}$$

This is equivalent to (191) in the unbounded case except when $\operatorname{int} P = \emptyset$; then, since $\bar{p} = 0$, $\bar{w}$ from (197) is undefined, but the solution to the SOCP problem (192) nevertheless exists.

The *minimax-rate* problem (200) differs from the SVM problem considered in e.g. [29] and [43] by the factor of $\frac{1}{||d_i||}$ included in each constraint. The difference is that whereas the standard SVM approach attempts to classify differences using a maximum-margin separating hyperplane, the minimax-rate approach finds the direction that maximizes the rate of constraint satisfaction; this is illustrated in Figure 30.

5.5 An Asymptotic Observer for Metric Cost Models

Suppose we have access to a very long (infinite) sequence of comparisons $S = \{(x_k^1, x_k^2)\}_{k=1}^{\infty} = \{s_1, s_2, \cdots\} \subset X \times X$, perhaps as the result of passive monitoring over an extended period of time, and we would like to know the features $\bar{x}$ of the ideal alternative. If alternatives are presented at random to the human, can we construct an asymptotic observer for $\bar{x}$ which can avoid storing all of the very (infinitely) many constraints implied by this sequence? It turns out that the answer is yes, and exactly such an observer is given by,

$$\tilde{x}_{k+1} = \begin{cases} P^k \tilde{x}_k + \frac{\alpha_k b_k}{d_k^T d_k} d_k & \text{if } d_k^T \tilde{x}_k - b_k > 0 \\ \tilde{x}_k & \text{otherwise} \end{cases} \tag{201}$$

$$P_k = I - \alpha^k \frac{d_k d_k^T}{d_k^T d_k} \tag{202}$$

for any sequence of observer gains $\alpha_k \in (0, 2)$ (and d_k, b_k defined by (164-166)), regardless of $\tilde{x}_0$. That is, $\tilde{x}_k$ converges to $\bar{x}$ in probability as $k \to \infty$, given a few

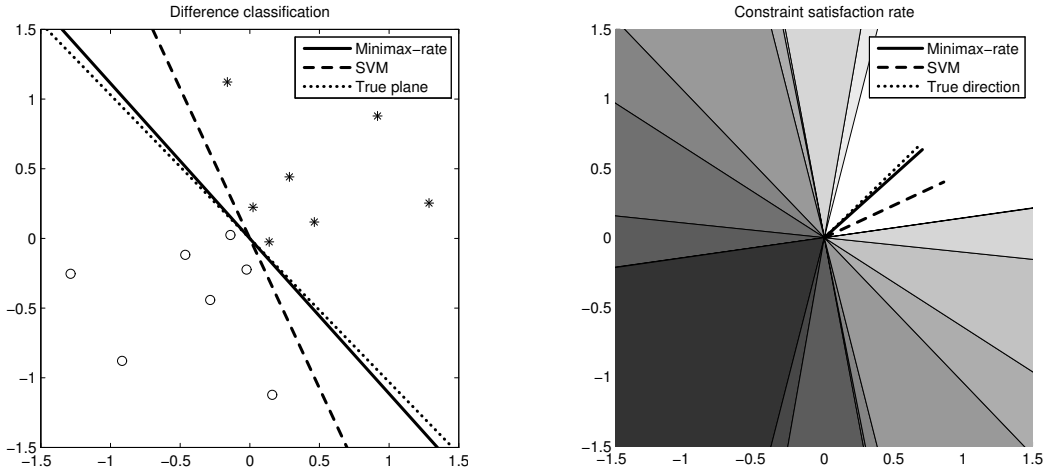

Figure 30: A number of uniformly-randomly selected points in $[-1,1] \times [-1,1] \subset \mathbb{R}^2$ are compared according to a point at infinity (i.e., a linear cost function) (dotted), and both the traditional SVM (dashed) and the minimax-rate (solid) approaches are used to produce estimates of this direction from the comparisons. From the difference-classification point of view (top), one wishes to separate the vectors $\{d_i\}_{i=1}^N$ (displayed as "o"s) from the vectors $\{-d_i\}_{i=1}^N$ (displayed as "*"s). From the minimax-rate point of view (bottom), one wishes to find the direction that maximizes the rate of constraint satisfaction (the numbers of violated constraints are represented by shades of gray; the white region is feasible). The traditional SVM solution separates the positive from the negative differences with a larger margin (top), but the minimax-rate solution stays as far from the edge of the constraint cone as possible (bottom).

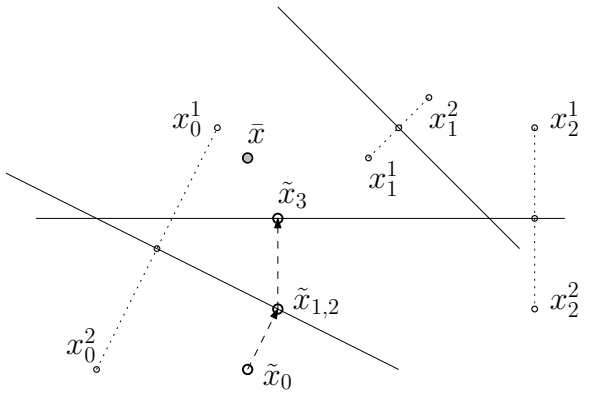

Figure 31: A series of the observer's estimates, with $\alpha_k = 1 \; \forall k$. The initial estimate is $\tilde{x}_0$, and the true ideal is given by $\bar{x}$. In step 0, the observer projects $\tilde{x}_0$ onto the plane (solid line) corresponding to the measured output $s_0 = (x_0^1, x_0^2)$ to produce $\tilde{x}_1$. In step 1, the observer makes no changes to its estimate, because $\tilde{x}_1$ is on the correct side of the plane corresponding to s_1; hence $\tilde{x}_2 = \tilde{x}_1$. In step 2, the observer projects $\tilde{x}_2$ onto the plane corresponding to s_2 to create the estimate $\tilde{x}_3$, which is yet closer to $\bar{x}$.

assumptions; we will prove this shortly in Theorem 16. Moreover, note that, although (201-202) are broken down into separate expressions for clarity of presentation, they are in fact all functions of $\tilde{x}^k$, so this observer can be implemented with only $\dim\{X\}$ real memory elements.

It should be noted that this observer can be viewed as a subgradient algorithm for solving the convex optimization problem; for more on subgradient algorithms, see e.g. [16].

Geometrically, the observer (201-202) operates through a series of projections (or under/over-projections, if $\alpha_k \neq 1$), as illustrated in Figure 31, with each projection bringing the estimate $\tilde{x}_k$ of the ideal closer to the true ideal, $\bar{x}$. A proof of convergence follows as Theorem 16.

Before continuing, we now state a useful lemma, whose geometric interpretation is that comparisons between distances relative to reference points can be interchanged with signed point-plane distance tests.

Lemma 5. *Let x^1, x^2, $\bar{x}$ be any vectors in an inner product space $(X, \langle \cdot, \cdot \rangle)$, and let*

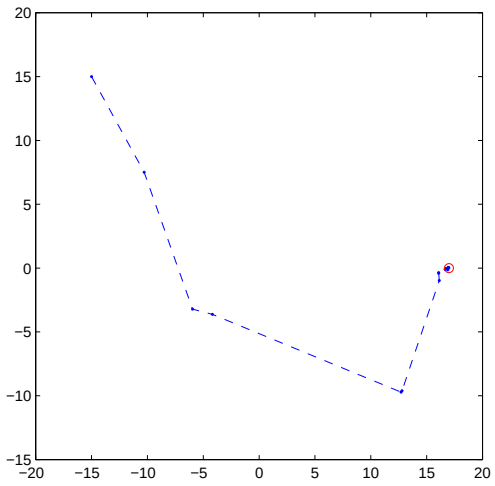

Figure 32: Example estimate trajectory for observer (201-202) for $\alpha^k = \alpha = 1$, with $X = \mathbb{R}^2$. The estimate begins at $\tilde{x}_0 = (-15, 15)$, and approaches the ideal $\bar{x} = (17, 0)$.

$\bowtie$ be a binary relation from the set, $\{=, <, >, \leq, \geq\}$. Then,

$$\bar{x} \in \{x \mid \langle d, x \rangle - b \bowtie 0\} \iff ||x^1 - \bar{x}|| \bowtie ||x^2 - \bar{x}||$$

where $d = x^2 - x^1$, and $b = \frac{1}{2}\langle d, x^1 + x^2 \rangle$.

The proof of this is based on the Polarization Identity and is straightforward.

Theorem 16. *Let $\bar{x} \in X$ be the ideal alternative, and $S = \{(x_k^1, x_k^2)\}_{k=1}^{\infty} = \{s_1, s_2, \cdots\}$ a sequence of pairs of i.i.d. random vectors drawn according to a probability density function p on $\{(x^1, x^2) \in X \mid ||x^1 - \bar{x}|| < ||x^2 - \bar{x}||\}$ which is nonzero in an open ball $B(\bar{x}, r) = B_r$ around $\bar{x}$. Then, the asymptotic observer given by (201-202) converges to $\bar{x}$ in probability.*

Proof : Please see the appendix.

An example of the estimate trajectory in feature space generated by such an observer is given in Figure 32. For this example, $X = \mathbb{R}^2$, and features were drawn from a uniform distribution in the square $[-20, 20] \times [-20, 20]$. The estimate evolves from its initial condition, $\tilde{x}_0 = (-15, 15)$ to near the ideal $\bar{x} = (17, 0)$.

Figure 33: Depicted are the 9 apples used to generate comparisons with the single orange.

5.6 Apples and Oranges

To demonstrate the application of these ideas, photos of nine apples were shown to an audience of thirteen people in a number of pairwise experiments. (The fruit is shown in Figure 33.)

Each apple was described by a 15-dimensional feature vector, containing (1-3) the average color in HSB (hue, saturation, brightness) color space, (4-6) the average color in RGB color space, (7) the color variance, (8-10) width, height, and the ratio of the two, (11-12) stem length, and angle relative to apple, (13-14) dimple angle and depth, and (15) roundness.

The partial order over the apples was thus generated by having a number of people make a number of randomly selected, pairwise comparisons (as the one depicted in Figure 34). Represented as a preference graph, the results of these experiments are given as Figure 35.

For these data, the minimization problem (173) is unbounded and hence we find

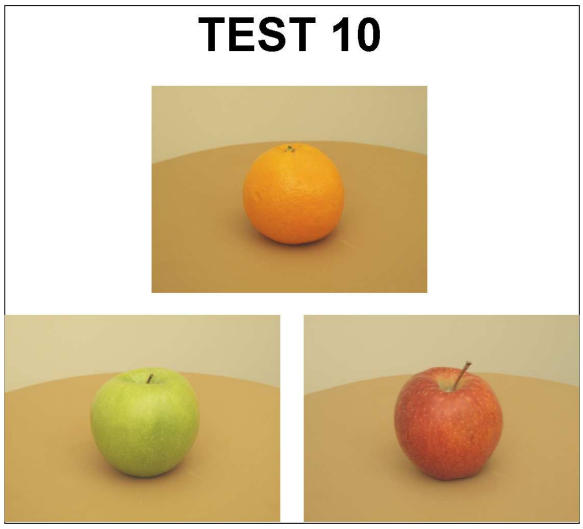

Figure 34: An example of a pairwise comparison between two apples, relative to the orange.

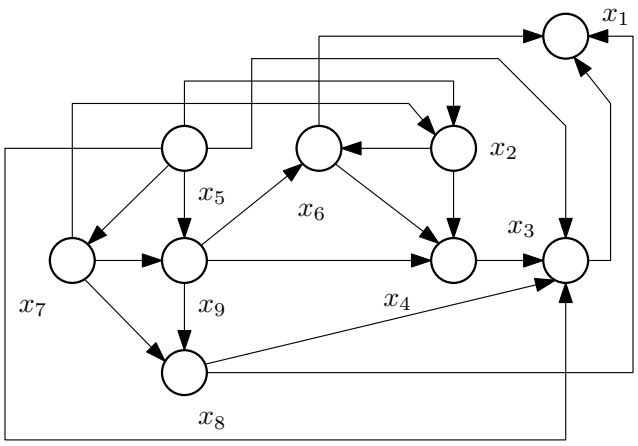

Figure 35: The preference graph corresponding to the apple experiments.

an optimal direction via (171). We obtain the optimum,

$$\begin{aligned}
\bar{v}_1 &= -0.0252 \quad \text{(Hue)} & \bar{v}_9 &= -0.2380 \quad \text{(Height)} \\
\bar{v}_2 &= -0.0844 \quad \text{(Saturation)} & \bar{v}_{10} &= -0.0472 \quad \text{(Width/Height)} \\
\bar{v}_3 &= 0.1374 \quad \text{(Brightness)} & \bar{v}_{11} &= -0.0409 \quad \text{(Stem Length)} \\
\bar{v}_4 &= 0.3572 \quad \text{(Red)} & \bar{v}_{12} &= -0.5017 \quad \text{(Stem Angle)} \\
\bar{v}_5 &= 0.1137 \quad \text{(Green)} & \bar{v}_{13} &= 0.6683 \quad \text{(Dimple Angle)} \\
\bar{v}_6 &= 0.1856 \quad \text{(Blue)} & \bar{v}_{14} &= 0.0996 \quad \text{(Dimple Depth)} \\
\bar{v}_7 &= 0.0442 \quad \text{(Variance)} & \bar{v}_{15} &= -0.0472 \quad \text{(Roundness)} \\
\bar{v}_8 &= -0.1593 \quad \text{(Width)}
\end{aligned}$$

which has the interpretation that dimple angle and redness are important orangelike qualities, and that large stem angles are perceived as un-orangelike.

5.7 Amoebas and Humans

To understand the comparison of higher-dimensional objects and in particular motions, another experiment was performed in which an audience of 25 people was asked to perform pairwise comparisons of different motions of a computer-animated amoeba, relative to the motion-captured movement of a human who danced the bhangra. An example of one such question is illustrated in Figure 37. In this manner, a preference graph was generated as before, with 12 vertices (the amoeba motions) and 20 edges; this is shown in Figure 36.

Inner products between the various amoeba motions were computed by rasterizing the motions to binary videos, blurring each frame of the result, and computing the standard Euclidean inner product of these (extremely large) `[Frame Width]`$\times$`[Frame Height]`$\times$`[Number of Frames]`-dimensional vectors. We note that the sheer size of this representation highlights the advantage of the instance vector expansion described in section 5.4.2, without which the optimization problem simply could not be realistically solved.

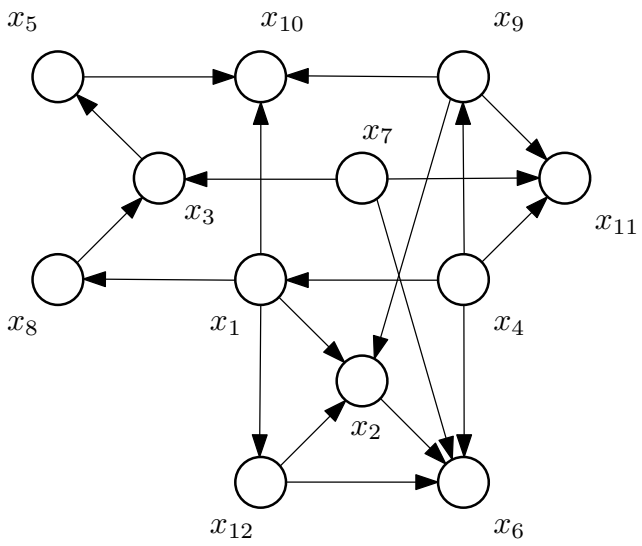

Figure 36: The preference graph corresponding to the amoeba experiments.

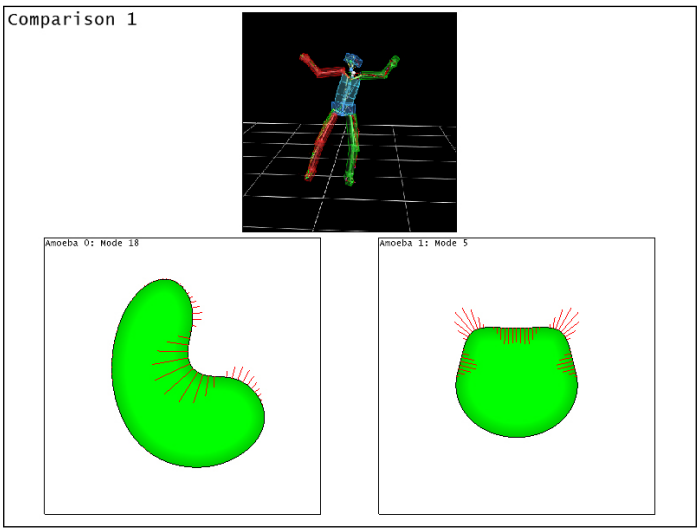

Figure 37: Each question took the form, "Which of the two 'amoebas' (bottom) looks more like the [motion capture data from a] human dancer (top)?"

The minimization problem (173) with the resulting data turns out to be unbounded and hence we again find an optimal direction via (171). We obtain the coefficient expansion for the optimal direction,

$$\bar{v} = \sum_{k=1}^{M} \xi_k x_k \tag{203}$$

where

$$\begin{aligned} \xi = 10^3(\ 1.4918, \quad -3.6556, \quad -0.1390, \quad 0.3113, \\ -1.1243, \quad -0.1771, \quad 2.6335, \quad 0.5878, \\ 1.8362, \quad -1.7319, \quad -0.2999, \quad 0.2672 \ ) \ . \end{aligned}$$

What this means is that, in order to look as much like it is dancing the bhangra as possible, an amoeba should as its first priority aspire to be as much like amoeba 7 ($\xi_7 = 2.6335$) and as dissimilar from amoeba 2 ($\xi_2 = -3.6556$) as possible, and that it should to a lesser extent model itself after amoebas 1 and 9 ($\xi_1 = 1.4918, \xi_9 = 1.8362$) while avoiding the aesthetically unappealing moves of amoebas 5 and 10 ($\xi_5 = -1.1243, \xi_{10} = -1.7319$). Although this does not explain why, psychologically, e.g. amoeba 7 is preferred to amoeba 2, it does produce both a consistent cost structure, and an estimate for an amoeba motion that will be preferred to all others in the larger space of motions.

5.8 Learning Metric Costs: Contributions

In this preliminary research, we investigated the problem of motion preference learning under the assumption of an underlying *metric cost model*; here, the alternatives being compared are points in a Hilbert space, and human judges are assumed to prefer one point to another if and only if it is closer to some fixed but unknown best alternative that they may not have been shown. This assumption appears to be a good one for the example considered and the features chosen, in that the feasible set P in this case is nonempty.

Based on the metric cost assumption, a Chebyshev estimator was given for the

best point for the case when P is bounded, and a natural generalization, the minimax-rate estimator, was developed for when P is unbounded. In the first case, the solution was found, with an efficiency rivaling standard quadratic SVMs, as the solution to a linear program; and in the second case the problem was shown to in fact reduce to a particular SVM classification problem. In addition, an asymptotic observer for metric cost functions was described, and shown to converge under reasonable assumptions.

In order that the estimators for the bounded and unbounded cases be applicable to situations in which the compared alternatives inhabit high- or infinite- dimensional metric spaces – as is the case for motion signals – the optimization problems were additionally given in an instance vector expansion form, which results in optimization problems whose size is proportional not to the dimensionality of the metric space, but only to the number of comparisons available.

In all cases, optimal cost functions *and* points/directions were found efficiently by convex programming. The results are efficient minimax estimators for the best possible alternative.

CHAPTER VI

CONCLUSIONS

We have addressed the coordination of multi-robot systems from two, physically-inspired points of view, and then produced algorithms that tune the resulting controllers to satisfy the subjective requirements of the humans who use them.

The first, *Eulerian*, family of controllers, uses tools from algebraic topology to create fluid-like flows, and is ultimately based on reinterpretations of discrete Laplacian operators not as communications protocols for mobile agents themselves, but rather as spatial operators. The spatial interpretation of discrete Laplacians parallels that of their continuous siblings, and is enabled by the existence, throughout developed environments, of static infrastructure with computing capacity. Discrete analogues to partial differential equations like the Laplace, Poisson, and Euler equations that occur throughout physics, naturally follow. These, in turn, provoke the investigation of other, novel Laplacian-like operators, whose null spaces are modified in various ways, to characterise different spaces of interest. Perhaps most importantly, all of the resulting algorithms are simple, linear, and globally-convergent – properties that are appealing both from an implementation and from an analysis point of view. The general mathematical approach employed here can be applied to a variety of problems, of which those discussed in this dissertation are but a small sampling.

The second, *Lagrangian*, family of controllers, using first geometric-mechanical formulations and then techniques from optimal control, produces algorithms that can be used to manipulate virtual structures of mobile robots built from pairwise constraints. More broadly, the optimal control algorithms described in this dissertation, although naturally applied to multiagent formations, are in fact developed in a way

that makes them generally applicable to nonlinear systems, and so have the potential for broad use. It shares with the first approach a geometric-topological inspiration – in this case, an interest in deforming space with homeomorphisms – as well as the use of simplicial complexes as spatial discretizations. What it does not share is the algorithmic simplicity, or the immediate mathematical richness (and tractability) of the Eulerian approach. Nevertheless, because it is directly compatible with existing approaches to formation control, the Lagrangian approach may, especially in the short term, find more application.

The final portion of this dissertation, which investigates the learning of human preferences to tune controllers, is less integrally related to multiagent coordination problems than are the previous two sections. It does, however, complement the development of Eulerian and Lagrangian control approaches, and helps to answer emerging questions about how humans can interact effectively with multiagent swarms. Additionally, it is related to the discrete Poisson-type equations of the Eulerian approach, so the two, apparently-disjoint areas, can and do cross-fertilize one another. These connections can only become stronger with future work.

APPENDIX A

PROOFS AND ADDITIONAL DISCUSSION

A.1 Proof of Theorem 12

Theorem. *The additional FONCs for the problem (138) are given by (139) and (139), which are reprinted here for convenience:*

$$\begin{aligned}\nabla_M \bar{J}(u,v,(M,z)) &= \int_0^T 2Q\left(M\bar{r}(\tau)+z-y(\tau)\right) r^T(\tau)d\tau \\ &\quad + \alpha\frac{2}{\det(M^TM)}\left(-\operatorname{tr}(M^TM)^pM^{-T}+p\operatorname{tr}(M^TM)^{p-1}M\right) \\ &\quad + 2\beta\frac{1}{p}\left(M-I\right) \\ &= \mathbf{0} \end{aligned} \tag{204}$$

$$\nabla_z \bar{J}(u,v,(M,z)) = \int_0^T 2Q(M\bar{r}(\tau)+z-y(\tau))d\tau + 2\gamma z = \mathbf{0} \tag{205}$$

which must be satisfied in addition to (103).

Before proving this, we present a few preliminaries with regard to notation. In what follows, $(x \mapsto [\text{expression}])$ denotes functions that take x as an argument and return [expression]; e.g., $(x \mapsto x^2)$ is the function that squares its argument. The notation $D_c(f)$ is used for the differential of a function f at a point c in the domain of f. Note that $D_c(f)$ is itself a (linear) function which can be evaluated; we denote its evaluation at h (the "direction" of variation of the argument to f) by $D_c(f)(h)$. For instance, $D_c(x \mapsto x^3)(z) = (h \mapsto 3c^2h)(z) = 3c^2z$. We denote the *gradient* of f at c by $\nabla f(c)$. The derivative is the vector *dual* to the gradient; that is $D_c(f)(h) = \langle \nabla f(c), h\rangle$ for all variations h of the argument to f, where $\langle\cdot,\cdot\rangle$ is an appropriate inner product. Continuing the preceding example, $\nabla(x \mapsto x^3)(c) = (x \mapsto 3x^2)(c) = 3c^2$.

We now give the proof of Theorem 12.

Proof : Considering the first term of (138),

$$D_A\left(M \mapsto \frac{\operatorname{tr}(M^TM)^p}{\det(M^TM)}\right)$$
$$= \operatorname{tr}(A^TA)^p D_A\left(M \mapsto \frac{1}{\det(M^TM)}\right) + \frac{1}{\det(A^TA)} D_A\left(M \mapsto \operatorname{tr}(M^TM)^p\right) \quad (206)$$

where, since

$$D_A\left(M \mapsto \det(M^TM)\right)$$
$$= \left(dX \mapsto \det(A^TA)\operatorname{tr}((A^TA)^{-1}dX)\right) \circ (dA \mapsto A^TdA + dA^TA)$$
$$= \left(dA \mapsto \det(A^TA)\operatorname{tr}((A^TA)^{-1}(A^TdA + dA^TA))\right)$$
$$= \left(dA \mapsto 2\det(A^TA)(\operatorname{tr}(A^{-1}dA)\right) \quad (207)$$

we have

$$D_A\left(M \mapsto \frac{1}{\det(M^TM)}\right)$$
$$= \left(dx \mapsto \frac{-dx}{\det(A^TA)^2}\right) \circ D_A\left(M \mapsto \det(M^TM)\right)$$
$$= \left(dA \mapsto \frac{-2\det(A^TA)\operatorname{tr}(A^{-1}dA)}{\det(A^TA)^2}\right)$$
$$= \left(dA \mapsto \frac{-2\operatorname{tr}(A^{-1}dA)}{\det(A^TA)}\right) \quad (208)$$

and

$$D_A\left(M \mapsto \operatorname{tr}(M^TM)^p\right)$$
$$= \left(dx \mapsto p\operatorname{tr}(A^TA)^{p-1}dx\right) \circ \left(dA \mapsto 2\operatorname{tr}(A^TdA)\right)$$
$$= \left(dA \mapsto 2p\operatorname{tr}(A^TA)^{p-1}\operatorname{tr}(A^TdA)\right) \quad (209)$$

Substituting (208) and (209) into (206),

$$\begin{aligned}
&D_A\left(M \mapsto \frac{\operatorname{tr}(M^TM)^p}{\det(M^TM)}\right) \\
&= \left(dA \mapsto -2\frac{\operatorname{tr}(A^TA)^p}{\det(A^TA)}\operatorname{tr}(A^{-1}dA) + \frac{2p\operatorname{tr}(A^TA)^{p-1}}{\det(A^TA)}\operatorname{tr}(A^TdA)\right) \\
&= \left(dA \mapsto \frac{-2}{\det(A^TA)}\operatorname{tr}\left(\left(\operatorname{tr}(A^TA)^pA^{-1} - p\operatorname{tr}(A^TA)^{p-1}A^T\right)dA\right)\right) \\
&= \left(dA \mapsto \frac{-2}{\det(A^TA)}\left\langle \operatorname{tr}(A^TA)^pA^{-T} - p\operatorname{tr}(A^TA)^{p-1}A, dA\right\rangle\right) \qquad (210)
\end{aligned}$$

where the angle brackets denote the inner product $\langle ., . \rangle$ defined for matrices by

$$\langle A, B\rangle = \operatorname{tr}(A^TB) \qquad (211)$$

for any two square matrices A and B of compatible dimensions.

This (210) is the dual to the gradient, which is given by (and replacing the symbol A by M),

$$-\frac{2}{\det(M^TM)}\left(\operatorname{tr}(M^TM)^pM^{-T} - p\operatorname{tr}(M^TM)^{p-1}M\right) \ . \qquad (212)$$

Likewise differentiating the other terms we obtain,

$$\begin{aligned}
\nabla_M\bar{J}(u,v,(M,z)) &= \int_0^T 2Q\left(M\bar{r}(\tau) + z - y(\tau)\right)r^T(\tau)d\tau \\
&+ \alpha\frac{2}{\det(M^TM)}\left(-\operatorname{tr}(M^TM)^pM^{-T} + p\operatorname{tr}(M^TM)^{p-1}M\right) + 2\beta\frac{1}{p}\left(M - I\right) = \mathbf{0}
\end{aligned} \qquad (213)$$

$$\nabla_z\bar{J}(u,v,(M,z)) = \int_0^T 2Q(M\bar{r}(\tau) + z - y(\tau))d\tau + 2\gamma z = \mathbf{0} \in \mathbb{R}^p \qquad (214)$$

which concludes the proof. ∎

A.2 Proof of Theorem 16

Theorem. *Let $\bar{x} \in X$ be the ideal alternative, and $S = \{(x_k^1, x_k^2)\}_{k=1}^{\infty} = \{s_1, s_2, \cdots\}$ a sequence of pairs of i.i.d. random vectors drawn according to a probability density function p on $\{(x^1, x^2) \in X \mid ||x^1 - \bar{x}|| < ||x^2 - \bar{x}||\}$ which is nonzero in an open ball*

$B(\bar{x}, r) = B_r$ around $\bar{x}$. Then, the asymptotic observer given by (201-202) converges to $\bar{x}$ in probability.

Proof : Please see the appendix. **1.** *If $\langle d_k, \tilde{x}_k \rangle - b_k > 0$, then $||\tilde{x}_{k+1} - \bar{x}|| < ||\tilde{x}_k - \bar{x}||$.* The distances $||\tilde{x}_k - \bar{x}||$ and $||\tilde{x}_{k+1} - \bar{x}||$ are related through the Polarization Identity by (where $\Delta_k = \tilde{x}_{k+1} - \tilde{x}_k$),

$$||\tilde{x}_{k+1} - \bar{x}||^2 = ||\tilde{x}_k + \Delta_k - \bar{x}||^2 =$$
$$||\tilde{x}_k - \bar{x}||^2 + ||\Delta_k||^2 + 2\langle \tilde{x}_k - \bar{x}, \Delta_k \rangle$$

so, it order to show that $||\tilde{x}_{k+1} - \bar{x}|| < ||\tilde{x}_k - \bar{x}||$, it is sufficient to demonstrate

$$||\Delta_k||^2 + 2\langle \tilde{x}_k - \bar{x}, \Delta_k \rangle < 0. \tag{215}$$

From (201, 202),

$$\begin{aligned} \Delta_k &= \left(I - \alpha^k \frac{d_k d_k^T}{d_k^T d_k} \right) \tilde{x}_k + \frac{\alpha_k b_k}{d_k^T d_k} d_k - \tilde{x}_k \\ &= \frac{\alpha}{\langle d_k, d_k \rangle} \left(b_k - \langle d_k, \tilde{x}_k \rangle \right) d_k \end{aligned} \tag{216}$$

so, substituting Δ_k into (215) (and dropping the superscript indices k),

$$\frac{\alpha^2}{\langle d, d \rangle} \left(b - \langle d, \tilde{x} \rangle \right)^2 + 2 \frac{\alpha}{\langle d, d \rangle} \left(b - \langle d, \tilde{x} \rangle \right) \langle d, \tilde{x} - \bar{x} \rangle \;\; < \;\; 0$$

or equivalently, so long as $\alpha > 0$ (as we require),

$$- \left(\langle d, \tilde{x} \rangle - b \right) \left[\alpha \left(b - \langle d, \tilde{x} \rangle \right) + 2 \langle d, \tilde{x} - \bar{x} \rangle \right] \;\; < \;\; 0. \tag{217}$$

Since by assumption $\langle d, \tilde{x} \rangle - b > 0$, this is satisfied iff the second factor is positive; that is,

$$\begin{aligned} \alpha \left(b - \langle d, \tilde{x} \rangle \right) + 2 \langle d, \tilde{x} - \bar{x} \rangle \;\; &= \\ \alpha b + (2 - \alpha) \langle d, \tilde{x} \rangle - 2 \langle d, \bar{x} \rangle \;\; &> \;\; 0. \end{aligned} \tag{218}$$

Since $\langle d, \tilde{x} \rangle > b$, and by Lemma 5, $\langle d, \bar{x} \rangle \leq b$, this is satisfied so long as $\alpha \in (0, 2)$, as we require.

2. *The sequence* $e_k = ||\tilde{x}_k - \bar{x}_k||,\ k = 0, 1, 2, ...$ *is nonincreasing.* In the second case of (201), $\tilde{x}_{k+1} = \tilde{x}_k$; this is nonincreasing. In the first case, $\langle d_k, \tilde{x}_k \rangle - b_k > 0$, so $e_{k+1} < e_k$ by point 1 above.

3. g.l.b.$(e_k) = 0$ *with unit probability.* By positivity of $||\cdot||$, zero is a lower bound. To show that this is the greatest such bound, consider some $\epsilon > 0$ and suppose that, at iteration m, $||\tilde{x}_m - \bar{x}|| = \epsilon$. Now, let $z = \min(r, \epsilon/2)$, and consider the open balls $B_1 = B(c_1, z/4), B_2 = B(c_2, z/4)$, where the center points c_1, c_2 are defined,

$$c_j = \bar{x} + \frac{\tilde{x} - \bar{x}}{||\tilde{x} - \bar{x}||} \frac{(2j-1)}{4} z$$

for $j \in \{1, 2\}$ (see Figure 38); additionally, let $x^1 \in B_1, x^2 \in B_2$. Then by Lemma 5, we can confirm that $\bar{x}$ and $\tilde{x}$ are on opposite sides of the plane corresponding to (x^1, x^2) (and hence, that a projection will occur) by verifying that,

$$||x^2 - \tilde{x}|| < ||x^1 - \tilde{x}|| \tag{219}$$

$$||x^2 - \bar{x}|| > ||x^1 - \bar{x}||. \tag{220}$$

Considering the first of these, we note by the triangle inequality,

$$||x^2 - \tilde{x}|| \leq ||x^2 - c_2|| + ||c_2 - \tilde{x}|| < \tfrac{1}{4}z + ||c_2 - \tilde{x}||$$

whereas, by the inverse triangle inequality,

$$\begin{aligned} ||x^1 - \tilde{x}|| &\geq |\ ||x^1 - c_1|| + ||c_1 - \tilde{x}||\ | \\ &\geq ||c_1 - \tilde{x}|| = \tfrac{1}{2}z + ||x^2 - c_2|| \end{aligned}$$

so this is indeed the case. Considering the second inequality (220), we have likewise,

$$||x^1 - \bar{x}|| \leq ||x^1 - c_1|| + ||c_1 - \bar{x}|| < \tfrac{1}{4}z + \tfrac{1}{4}z = \tfrac{1}{2}z$$

and

$$||x^2 - \bar{x}|| \geq |\ ||x^2 - c_2|| - ||c_2 - \bar{x}||\ | \geq \tfrac{3}{4}z$$

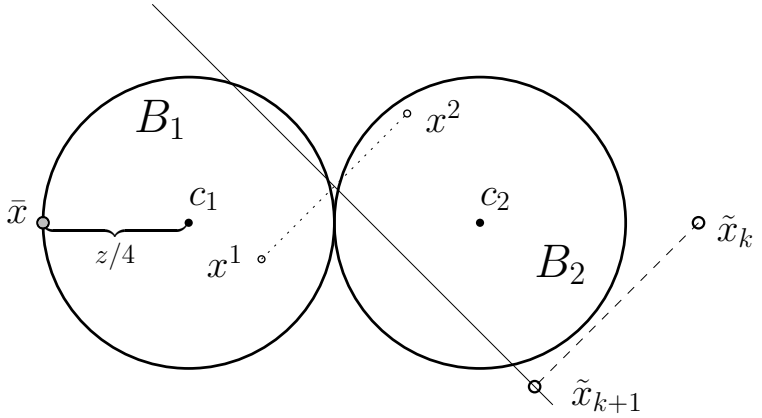

Figure 38: If $x^1 \in B_1$ and $x^2 \in B_2$, then $||\tilde{x}_{k+1} - \bar{x}|| < ||\tilde{x}_k - \bar{x}||$.

so this inequality holds as well. Therefore, *any* x^1, x^2 from B_1, B_2 are associated with a plane that separates $\tilde{x}$ from $\bar{x}$ and hence triggers a projection. Since B_1 and B_2 have nonzero measure, and are subsets of B_r in which $p(\cdot)$ is nonzero, then the probabilities for this iteration $P_1 = \mathrm{P}(\text{“a point is selected in } B_1\text{”})$ and $P_2 = \mathrm{P}(\text{“a point is selected in } B_2\text{”})$ are both nonzero, and therefore, since the s^k are independent, $P_{\text{both}} = \mathrm{P}(\text{“one point is selected in } B_1 \text{ and the other is selected in } B_2\text{”}) = P_1 P_2$ is nonzero, and the probability that this occurs for *at least* one iteration $k > m$ is given by $1 - \prod_{k=m}^{\infty} \left(1 - P_{\text{both}}^k\right) = 1$ or in other words, with probability one, there exists a $q > m$ such that $\langle d_q, \tilde{x}_q \rangle - b_q > 0$. Then, by point 1, $||\tilde{x}_q - \bar{x}|| < ||\tilde{x}_m - \bar{x}|| = \epsilon$, and so ϵ, with unit probability, cannot be a lower bound. Since e_k is a nonincreasing sequence in $\mathbb{R}$ and g.l.b.$(e_k) = 0$, e_k converges to 0 and thus $\tilde{x}$ converges to $\bar{x}$ in probability. ∎

A.3 Relationship of Metric Cost Models to SVM Approaches in the Bounded Case

The metric cost models considered in the previous section are a special case of quadratic cost models, subject to additional constraints. We will discuss their relationship via kernelization to linear cost models in this section.

A.3.1 Quadratic Cost Models

First, assume for notational convenience that X is finite-dimensional, and consider an arbitrary quadratic cost of the form

$$f(x) = x^T Q x + b^T x \tag{221}$$

along with the problem of picking Q and b such that f is maximum-margin. We immediately note that (221) can be written as an inner product in $\mathbb{R}^{\dim(X)\times\dim(X)} \times \mathbb{R}^{\dim(X)}$ as

$$f(x) = \left\langle (Q, b), (xx^T, x) \right\rangle = \langle (Q, b), \Phi(x) \rangle \tag{222}$$

where $\Phi : X \to \mathbb{R}^{\dim(X)\times\dim(X)} \times \mathbb{R}^{\dim(X)}$ is defined by the above, and in this space the inner product is defined

$$\langle (Q_1, b_1), (Q_2, b_2) \rangle = \langle Q_1, Q_2 \rangle + \langle b_1, b_2 \rangle = \operatorname{tr} Q_1^T Q_2 + b_1^T b_2. \tag{223}$$

I.e., this is the inner product induced on the product space by the Frobenius inner product on $\mathbb{R}^{\dim(X)\times\dim(X)}$ and the Euclidean inner product on $\mathbb{R}^{\dim(X)}$.

The kernel corresponding to Φ is,

$$\begin{aligned}
\kappa(x, y) &= \langle \Phi(x), \Phi(y) \rangle \\
&= \left\langle (xx^T, x), (yy^T, y) \right\rangle \\
&= \operatorname{tr}(x(x^T y)y^T) + x^T y \\
&= (x^T y) \operatorname{tr}(xy^T) + x^T y \\
&= (x^T y)^2 + x^T y \\
&= \langle x, y \rangle^2 + \langle x, y \rangle \ .
\end{aligned} \tag{224}$$

Hence, if no additional constraints are placed on (Q, b), then the quadratic case is reduced by a simple example of kernelization to the linear case. In the subsections that follow we examine what happens when additional constraints are imposed.

A.3.2 Metric Cost Models: Known Metric

By the Polarization Identity metric costs can be rewritten,

$$f(x) = ||x||^2 - 2\langle \bar{x}, x \rangle + ||\bar{x}||^2 \tag{225}$$

or, dropping the $||\bar{x}||^2$ term since it is constant in x,

$$\begin{aligned} f(x) &= ||x||^2 - 2\langle \bar{x}, x \rangle \\ &= \left\langle (I, \bar{x}), (xx^T, -2x) \right\rangle \\ &= \langle (I, \bar{x}), \Phi(x) \rangle \end{aligned} \tag{226}$$

where Φ is defined by the above, and has corresponding kernel

$$\kappa(x, y) = \langle x, y \rangle^2 + 4\langle x, y \rangle \ . \tag{227}$$

One might be tempted to try solving the original optimization problem (157) over (Q, b) as in the previous subsection using this kernel, now subject to the equality constraint that $Q = I$. Then we could recover the optimal point as $\bar{x} = -\frac{1}{2}b$. Unfortunately, adding this equality constraint ruins the important property of the optimization problem that the optimal (Q, b) lies in the span of the (transformed) data and so that the cost at a point can be reconstructed as a linear combination of the kernel evaluated at support vectors and the point. Although the problem is still convex, its dimensionality is that of X. In the infinite- or very-large- dimensional case, this renders the resulting optimization problem impractically large. Hence, standard Support Vector Machines are incompatible with metric cost models, which helps to motivate the preceding Chebyshev estimation scheme.

REFERENCES

[1] Adomavicius, G. and Tuzhilin, A., "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. on Knowl. and Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.

[2] Afriat, S. N., "The construction of utility functions from expenditure data," *International Economic Review*, vol. 8, pp. 67–77, Feb 1967.

[3] Agarwal, S., "Ranking on graph data," in *In ICML*, pp. 25–32, ACM Press, 2006.

[4] Aguiar, A. P., Hespanha, J. P., and Kokotović, P. V., "Performance limitations in reference-tracking and path-following for nonlinear systems," *Automatica*, vol. 44, pp. 598–610, Mar. 2008.

[5] Aguiar, A. and Hespanha, J., "Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty," in *Automatic Control, IEEE Transactions on*, vol. 52, pp. 1362–1379, 2007.

[6] Aho, A. V., Garey, M. R., and Ullman, J. D., "The transitive reduction of a directed graph," *SIAM Journal on Computing*, vol. 1, pp. 131–137, Jun 1972.

[7] Aiolli, F. and Sperduti, A., "Learning preferences for multiclass problems," in *Advances in Neural Information Processing Systems 17*, pp. 17–24, MIT Press, 2004.

[8] Aizerman, M., Braverman, E., and Rozonoer, L., "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Remote Control*, vol. 25, pp. 821–837, 1964.

[9] Akleman, E., "Making caricatures with morphing," in *SIGGRAPH: ACM Special Interest Group on Computer Graphics and Interactive Techniques*, p. 145, 1997.

[10] Bahamonde, A., Diez, J., Quevedo, J., Luances, O., and del Coz, J., "How to learn consumer preferences from the analysis of sensory data by means of support vector machines (svm)," *Trends in Food Science and Technology*, vol. 18, pp. 20–28, 2007.

[11] Balch, T., Arkin, R. C., and Member, S., "Behavior-based formation control for multi-robot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 926–939, 1999.

[12] Beard, R., Lawton, J., and Hadaegh, F. vol. 6, pp. 4087 – 4091, 2000.

[13] Bodnar, M. and Velazquez, J. J. L., "Derivation of macroscopic equations for individual cell-based models. a formal approach.," *Mathematical Methods in the Applied Sciences*, vol. 28, pp. 1757–1779, 2005.

[14] Borges, L. and Oliveira, S., "A parallel davidson-type algorithm for several eigenvalues," vol. 144, pp. 727–748, 1998.

[15] Boser, B. E., Guyon, I. M., and Vapnik, V. N., "A training algorithm for optimal margin classifiers," in *5th Annual ACM Workshop on COLT* (Haussler, D., ed.), pp. 144–152, ACM Press, 1992.

[16] Boyd, S. and Vandenberghe, L., *Convex Optimization*. Cambridge University Press, 2004.

[17] Bradley, R. A. and Terry, M. E., "Rank analysis of incomplete block designs: The method of paired comparisons," *Biometrica*, vol. 39, pp. 324–345, 1952.

[18] Bullo, F. and Lewis, A. D., *Geometric Control of Mechanical Systems: Modeling, Analysis, and Design for Simple Mechanical Control Systems*. Springer, 2004.

[19] Burger, M., Capasso, V., and Morale, D., "On an aggregation model with long and short range interactions," *Nonlinear Analysis: Real World Applications*, vol. 8, no. 3, pp. 939 – 958, 2007.

[20] C. Myers, L.R. Rabiner, A. R., "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," in *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-6, pp. 623–635, Dec 1980.

[21] Carr, H., Möller, T., and Snoeyink, J., "Artifacts caused by simplicial subdivision," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 231–242, 2006.

[22] Chu, W. and Ghahramani, Z., "Preference learning with gaussian processes," in *ICML '05: Proceedings of the 22nd international conference on Machine learning*, (New York, NY, USA), pp. 137–144, ACM, 2005.

[23] Chu, W. and Keerthi, S. S., "Support vector ordinal regression," *Neural Computation*, vol. 19, p. 2007, 2007.

[24] Cohen, W. W., Schapire, R. E., and Singer, Y., "Learning to order things," *Journal of Artificial Intelligence Research*, vol. 10, pp. 243–270, 1999.

[25] Coppersmith, D. and Winograd, S., "Matrix multiplication via arithmetic progressions," in *STOC '87: Proceedings of the nineteenth annual ACM symposium on Theory of computing*, (New York, NY, USA), pp. 1–6, ACM, 1987.

[26] Cortes, C., Mohri, M., and Rastogi, A., "Magnitude-preserving ranking algorithms," in *Proceedings of the Twenty-fourth International Conference on Machine Learning (ICML 2007)*, pp. 169–176, 2007.

[27] Desbrun, M., Hirani, A., Leok, M., and Marsden, J., "Discrete exterior calculus." (preprint, `http://arxiv.org/abs/math/0508341`), 2003.

[28] Diestel, R., *Graph Theory*. Springer, 2005.

[29] Díez, J., del Coz, J. J., Luaces, O., and Bahamonde, A., "Clustering people according to their preference criteria," *Expert Syst. Appl.*, vol. 34, no. 2, pp. 1274–1284, 2008.

[30] Do, K. and Pan, J., "Nonlinear formation control of unicycle-type mobile robots," *Robotics and Autonomous Systems*, vol. 55, no. 3, pp. 191 – 204, 2007.

[31] Egerstedt, M. and Hu, X., "Formation constrained multi-agent control," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 947–951, December 2001.

[32] Egerstedt, M. and Hu, X., "Formation constrained multi-agent control," in *Proc. ICRA Robotics and Automation IEEE Int. Conf*, vol. 4, pp. 3961–3966, 2001.

[33] Evans, L. C., *Partial Differential Equations*. American Mathematical Society, June 1998.

[34] Fiechter, C.-N. and Rogers, S., "Learning subjective functions with large margins," in *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 287–294, Morgan Kaufmann, 2000.

[35] Fischer, M. J. and Meyer, A. R., "Boolean matrix multiplication and transitive closure," in *Twelfth Annual Symposium on Switching and Automata Theory*, pp. 129–131, 1971.

[36] G. Ferrari-Trecate, A. B. and Gati, M., "Analysis of coordination in multi-agent systems through partial difference equations. part I: The Laplacian control.," in *16th IFAC World Congress on Automatic Control*, 2005.

[37] Grady, L. J. and Polimeni, J. R., *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. Springer, 2010.

[38] Guillotin-Plantard, N. and Schott, R., *Dynamic Random Walks*, p. 241. Elsevier, 2006.

[39] Guldner, J., Utkin, V., Hashimoto, H., and Harashima, F., "Tracking gradients of artificial potential fields with non-holonomic mobile robots," in *Proc. American Control Conference*, vol. 4, pp. 2803–2804, 1995.

[40] Hatcher, A., *Algebraic Topology*. Cambridge University Press, 2002. http://www.math.cornell.edu/~hatcher/AT/ATpage.html.

[41] Hauser, J. and Hindman, R., "Aggressive flight maneuvers," in *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, vol. 5, pp. 4186–4191, 1997.

[42] Herbrich, R., Graepel, T., Bollmann-Sdorra, P., and Obermayer, K., "Supervised learning of preference relations," in *Proceedings FGML-98, German National Workshop on Machine Learning*, pp. 43–47, 1998.

[43] Herbrich, R., Graepel, T., and Obermayer, K., "Large margin rank boundaries for ordinal regression," in *Advances in large margin classifiers*, pp. 115–132, MIT Press, 2000.

[44] Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T., "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[45] Herodotou, N. and Venetsanopoulos, A. N., "Temporal prediction of video sequences using an image warping technique based on color segmentation," in *Image Analysis and Processing*, pp. 494–501, April 2006.

[46] Holzrichter, M. and Oliveira, S., "A graph based method for generating the fiedler vector of irregular problems," in *In Lecture Notes in Computer Science*, pp. 978–985, Lecture, 1999.

[47] Hörmander, L., *The Analysis of Linear Partial Difference Operators*, vol. 1. Springer-Verlag, 1983.

[48] Hüllermeier, E., Frnkranz, J., Cheng, W., and Brinker, K., "Label ranking by learning pairwise preferences," *Artificial Intelligence*, vol. 172, pp. 1897–1917, 2008.

[49] Ji, M., Azuma, S., and Egerstedt, M., "Role-assignment in multi-agent coordination," *International Journal of Assistive Robotics and Mechatronics*, vol. 7, pp. 32–40, March 2006.

[50] Jiang, X., Lim, L.-H., Yao, Y., and Ye, Y., "Statistical ranking and combinatorial hodge theory," *Mathematical Programming*, vol. 127, pp. 203–244, 2011.

[51] Joachims, T., "Optimizing search engines using clickthrough data," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 133–142, 2002.

[52] Johnson, E. and Murphey, T., "Dynamic modeling and motion planning for marionettes: Rigid bodies articulated by massless strings," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 330–335, 2007.

[53] Kemeny, J. G. and Snell, L., "Preference ranking: an axiomatic approach," pp. 9–23, 1973.

[54] Keysers, D. and Unger, W., "Elastic image matching is NP-complete," *Pattern Recognition Letters*, vol. 24, pp. 445–453, 2003.

[55] Kingston, P. and Egerstedt, M., "Distributed-infrastructure multi-robot routing using a helmholtz-hodge decomposition," in *Proc. Conference on Decision and Control*, pp. 5281–5286, 2011.

[56] Kingston, P., Egerstedt, M., and Verriest, E., "Health monitoring of networked systems," in *Proc. Mathematical Theory of Networks and Systems*, 2008.

[57] Kingston, P. and Egerstedt, M., "Metric preference learning with applications to motion imitation," *SIAM J. Control and Optimization*. Submitted.

[58] Kingston, P. and Egerstedt, M., "Comparing apples and oranges through partial orders: An empirical approach," in *American Control Conference*, pp. 5434–5439, 2009.

[59] Kingston, P. and Egerstedt, M., "Time and output warping of control systems," in *American Control Conference*, pp. 6042–6047, 2010.

[60] Kingston, P. and Egerstedt, M., "Motion preference learning," in *American Control Conference*, pp. 3819–3824, 2011.

[61] Kingston, P. and Egerstedt, M., "Time and output warping of control systems: Comparing and imitating motions," *Automatica*, vol. 47, pp. 1580–1588, 2011.

[62] Kloder, S., Bhattacharya, S., and Hutchinson, S., "A configuration space for permutation-invariant multi-robot formations," in *Proc. IEEE Int'l. Conf. on Robotics and Automation*, pp. 2746–2751, 2004.

[63] Kloder, S. and Hutchinson, S., "Path planning for permutation-invariant multi-robot formations," in *Proc. IEEE Int'l. Conf. on Robotics and Automation*, pp. 1797–1802, 2005.

[64] Korada, S., Montanari, A., and Oh, S., "Gossip pca," 2011.

[65] Kunz, T., Kingston, P., Stilman, M., and Egerstedt, M., "Dynamic chess: Strategic planning for robot motion," in *Proc. Int'l. Conference on Robotics and Automation (ICRA)*, pp. 3796–3803, 2011.

[66] Kwakernaak, H. and Sivan, R., "The maximal achievable accuracy of linear optimal regulators and linear optimal filters.," *IEEE Trans. on Automat. Contr.*, vol. 17, no. 1, pp. 79–86, 1972.

[67] Kybic, J., *Elastic Image Registration using Parametric Deformation Models*. Phd in biomedical image processing, Ecole Polytechnique Fédérale de Lausanne, July 2001.

[68] Laurent, T., "Local and global existence for an aggregation equation," *Communications in Partial Differential Equations*, vol. 32, pp. 1941–1964, December 2007.

[69] Lee, J. M., *Introduction to Smooth Manifolds*. Springer, 2000.

[70] Lei, H. and Govindaraju, V., "Direct image matching by dynamic warping," in *Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, p. 76, 2004.

[71] Leonard, N. E. and Fiorelli, E., "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. 40th IEEE Conference on Decision and Control*, pp. 2968–2973, December 2001.

[72] Mesbahi, M. and Egerstedt, M., *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[73] Mogilner, A. and Edelstein-Keshet, L., "A non-local model for a swarm," *Journal of Mathematical Biology*, vol. 38, pp. 534–570, 1999.

[74] Montaner, M., López, B., and De La Rosa, J. L., "A taxonomy of recommender agents on the internet," *Artif. Intell. Rev.*, vol. 19, no. 4, pp. 285–330, 2003.

[75] Muhammad, A. and Jadbabaie, A., "Decentralized computation of homology groups in networks by gossip," in *Proc. American Control Conference*, pp. 3438–3443, 2007.

[76] Munkres, J. R., *Elements of Algebraic Topology*. Addison-Wesley, 1984.

[77] Munro, I., "Efficient determination of the strongly connected components and the transitive closure of a graph." Unpublished manuscript, 1971.

[78] Olfati-Saber, R. and Murray, R. M., "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[79] Olfati-Saber, R. and Murray, R. M., "Distributed cooperative control of multiple vehicle formations using structural potential functions," in *IFAC World Congress*, pp. 346–352, 2002.

[80] Pappas, G., "Avoiding saturation by trajectory reparameterization," in *Proceedings of the 35th IEEE Decision and Control*, vol. 1, pp. 76–81, Dec 1996.

[81] Perot, J. B. and Subramanian, V., "Discrete calculus methods for diffusion," *J. Comput. Phys.*, vol. 224, pp. 59–81, May 2007.

[82] Prochazka, Z., Ito, T., and Okamoto, T., "Image correspondences by warping functions and its applications," in *Systems and Computers in Japan*, vol. 33, pp. 22–30, 2002.

[83] Raptis, M., Bustreo, M., and Soatto, S., "Time warping under dynamic constraints," in *Eleventh IEEE International Conference on Computer Vision, Workshop on Dynamical Vision*, 2007.

[84] Restificar, A. and Haddawy, P., "Inferring implicit preferences from negotiation actions," in *Proc. Int'l Symposium on Artificial Intelligence and Mathematics*, Jan 2004.

[85] Reynolds, C. W., "Flocks, herds, and schools: A distributed behavioral model," in *Computer Graphics*, pp. 25–34, 1987.

[86] Sakoe, H. and Chiba, S., "A dynamic programming approach to continuous speech recognition," in *Proc. Int'l. Cong. Acoust.*, pp. 65–68, 1971.

[87] Sakoe, H. and Chiba, S., "Dynamic programming algorithm optimization for spoken word recognition," in *IEEE Transactions on Acoustics, Speech and Signal Processing*, pp. 43–49, Feb 1978.

[88] Skjetne, R., Fossen, T. I., and Kokotovic, P. V., "Adaptive maneuvering, with experiments, for a model ship in a marine control laboratory," in *Automatica*, vol. 41, pp. 289–298, 2005.

[89] Smola, A. J. and Schölkopf, B., "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[90] Srinivasan, A. and Mascagni, M., *Monte Carlo Techniques for Estimating the Fiedler Vector in Graph Applications,*. 2002.

[91] Strassen, V., "Gaussian elimination is not optimal," *Numer. Math*, vol. 13, no. 3, pp. 354–356, 1969.

[92] Tahbaz-Salehi, A. and Jadbabaie, A., "Distributed coverage verification algorithms in sensor networks without location information," *IEEE Transactions on Automatic Control*, vol. 55, pp. 1837–1849, August 2010.

[93] Takaya, K. and Reinhardt, R. T., "Low bit-rate facial motion picture coding using image warping," in *WESCANEX 97: Communications, Power and Computing. Conference Proceedings., IEEE*, pp. 138–143, May 1997.

[94] Tan, K.-H. and Lewis, M. A., "Virtual structures for high-precision cooperative mobile robotic control," in *Proc. IEEE/RSJ Int Intelligent Robots and Systems '96, IROS 96 Conf*, vol. 1, pp. 132–139, 1996.

[95] Tanner, H. G., Jadbabaie, A., and Pappas, G. J., "Stable flocking of mobile agents, part i: Fixed topology," in *IEEE Conference on Decision and Control*, pp. 2010–2015, 2003.

[96] Tanner, H. G., Jadbabaie, A., and Pappas, G. J., "Stable flocking of mobile agents, part ii: Dynamic topology," in *IEEE Conference on Decision and Control*, pp. 2016–2021, 2003.

[97] Thurstone, L. L., *The measurement of values*. University of Chicago Press, 1959.

[98] Tong, Y., Lombeyda, S., Hirani, A. N., and Desbrun, M., "Discrete multiscale vector field decomposition," in *Proc. SIGGRAPH*, 2003.

[99] Topaz, C. M., Bertozzi, A. L., and Lewis, M. A., "A nonlocal continuum model for biological aggregation," *Bulletin of Mathematical Biology*, vol. 68, pp. 1601–1623, October 2006.

[100] Twu, P. and Egerstedt, M., "Controllability of homogeneous single-leader networks," in *49th IEEE Conference on Decision and Control*, December 2010.

[101] Uchida, S. and Sakoe, H., "Piecewise linear two-dimensional warping," in *Systems and Computers in Japan*, vol. 32, pp. 1–9, 2001.

[102] Vapnik, V. N., *The nature of statistical learning theory*. Springer-Verlag, 1995.

[103] Varian, H. R., "The nonparametric approach to demand analysis," *Econometria*, vol. 50, pp. 945–973, Jul 1982.

[104] Vidal, R., Shakernia, O., and Sastry, S., "Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation," in *In IEEE International Conference on Robotics and Automation*, pp. 584–589, 2003.

[105] Waibel, A. and Yegnanarayana, "Comparative study of nonlinear time warping techniques in isolated word speech recognition," in *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 31, pp. 1582–1586, Feb 1978.

[106] Wang, P. K. C. and Hadaegh, F. Y., "Coordination and control of multiple microspacecraft moving in formation," 1996.

[107] Xiao, L. and Boyd, S., "Fast linear iterations for distributed averaging," in *Proc. 42nd IEEE Conf. Decision and Control*, vol. 5, pp. 4997–5002, 2003.

[108] Yamaguchi, H. and Burdick, J. W., "Asymptotic stabilization of multiple nonholonomic mobile robots forming group formations," in *Proc. IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3573–3580, May 1998.

[109] Zavlanos, M. M. and Pappas, G. J., "Distributed formation control with permutation symmetries," in *Proc. IEEE Conference on Decision and Control*, December 2007.

[110] Zelazo, D. and Mesbahi, M., "Edge agreement: Graph-theoretic performance bounds and passivity analysis," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 544–555, 2011.